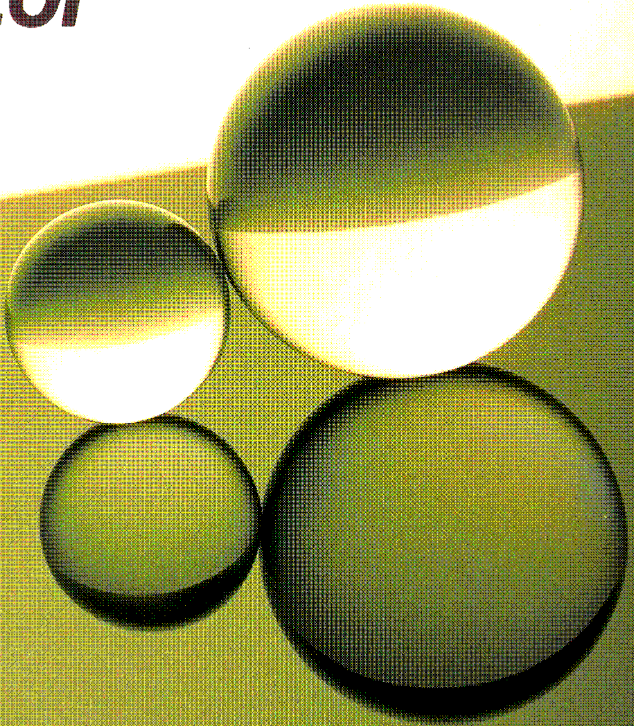
Vz Editor

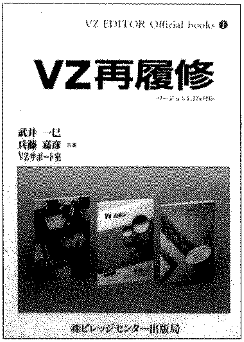


Version 1.5

- [ ] 5インチ
- [x] 3.5インチ

NEC 9800シリーズ
EPSON286シリーズ
J3100、IBM PC、AX

定価 9,800円



ISBN4-938704-11-4

VZ EDITOR Official Books ①

VZ再履修 Ver.1.57a対応

武井一巳・兵藤嘉彦 共著
A5判288頁
定価1,500円(本体1,456円)

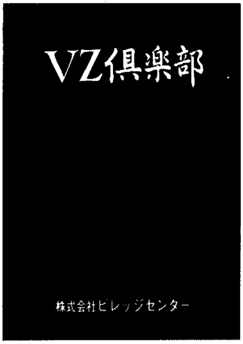
マニュアルがちょっと難しいと思っている貴方の悩みを解消する1冊。武井一巳氏によるインストールから使い方の初歩までの易しい解説にくわえ、兵藤嘉彦氏が回答するQ＆Aも収録。

VZ天國

VZエディタ作者 兵藤 嘉彦監修
A5判288頁(3.5"＆5"FD付き)
定価4,500円(本体4,368円)

VZエディタのマクロの魅力を満載したVZオフィシャルブック。実用的なツール類をはじめ、ドラクエ風RPG I＆II、ブロック崩しなどのゲームも収録。VZマクロプログラマ座右の書であるとともにすべてのVZユーザにとって必読の書です。

ISBN4-938704-06-4



VZ倶楽部

ビレッジセンター編
A5判 392頁 5"FD付き
定価3,800円(本体3,689円)

VZユーザが、それぞれ思いのままにつづった文章をこの一冊にまとめました。付録FDには、VZエディタ Ver.1.5から1.56の差分ファイル、テキスト印刷ソフトPRT++、EZ9801(VZの前身EZエディタ)を収録しました。

ビレッジセンター出版案内

ビレッジセンターの書籍は全国の書店にてお求めになれます。

VZ Editor Version 1.5

ユーザーズマニュアル

Copyright (C) 1989 by c.mos

Published by Village Center

はじめに

このたびは、VZエディタをお買い求め頂きまして、誠にありがとうございます。

VZエディタ（略してVZ）は、今後、MS-DOSマシンの標準エディタとなるべく、十分な性能と操作性および柔軟性をもった高機能ツールです。その設計思想は、ユーザー一人一人の嗜好に合せて、最大限の使い易さを実現する事にあります。そのため、他に類をみないカスタマイズ機能の充実と、強力なマクロ機能を有するに至りました。どうか皆様方が、本マニュアルを御詳読の上、それぞれ御自分に合ったVZで、最高の操作性を体験される事を願います。
Ver1.5では、EMSの対応等、操作環境をより充実するとともに、J-3100版、IBM PC／AX版と対応機種を広げました。

ここで改めて、VZの開発に際し、甚大なる御協力を賜わったモニタユーザーの方々に、心から感謝いたします。

1989年12月　　兵藤嘉彦（c.mos）

本ソフトウェアのご使用について

1. 本ソフトウェアの著作権は、兵藤嘉彦（c.mos）が所有しています。また販売権は、Village Center Inc.が所有しています。
2. 本ソフトウェア（VZエディタ）またはその複製を、同時に2人以上で使用することはできません。
3. 本ソフトウェアには、ソースコードが付属しています。本製品を改良もしくは他機種へ移植する場合に限り、その変更・追加された部分のソースリストの転載・頒布は自由です。
4. 本ソフトウェアに含まれるファイルのうち、VZエディタ本体（vz???.com）とそのソースコードを除くすべてのファイルの転載・頒布は自由です。
5. 3.4項の転載・頒布に際しては、1項で述べた所有権を明示しなくてはなりません。
6. 本マニュアルの一部または全部を、販売者に無断で複写・転載することはできません。
7. 本製品をご使用になった上で生じたいかなる損害に対しても、著作者および販売者は一切の責任を負いません。
8. 学校や企業において多人数でのご購入を予定される場合は、ご相談ください。

アフターサポートについて

■ アフターサポート

アフターサポートや将来のバージョンアップをご希望の方は、事前に弊社までレジストレーションカードをご返送ください。

また、シリアルナンバーは、本マニュアルの表紙を開けたところに差込み製本してあります。このシリアルナンバーは、レジストレーションカードと、控え部分以外には記入してありませんので、本マニュアルは、大切に保管してください。

■ テクニカルサポート

テクニカルサポートの窓口は、下記のとおりです。

・サポート専用電話　0424-88-8752

（祭日を除く、毎週（火）（金）の 13：00～16：00）

・FAX　0424-88-5471

※　これ以外のサポート（郵便等）は原則的に受け付けておりません。

※　誤動作が発生し、サポートをお受けになる場合は、事前に以下の事項をチェックしてください。

・誤動作の再現性はあるか？

・標準添付のDEFファイルで再現するか？

・他の常駐ソフトやFEP等を外しても再現するか？

・MS-DOSのバージョンが古くないか？

※　本Ver1.57は、Ver1.5の最終バージョンです。

次のVer2.0の発売時期は未定ですが、バージョンアップの際は、必ずご案内をお送りします。購入後は、お早めに登録葉書をご投函ください。

ユーザ登録完了のご連絡は、いたしておりませんのでご了解ください。

また、住所の変更等がございましたら、葉書あるいはファクシミリにて、ビレッジセンター本部へご連絡ください。

※　本ディスクに含まれる VZIBM.COM は、英語版DOS用であり、DOS／Vでは動作いたしません。

DOS／V、PS／55版VZを別途用意しておりますので、お求めになりたい方は、マニュアル巻末のDOS／V申込専用郵便振替用紙にてお申し込みください。

※　VZの最新情報は、日経MIXの v.c. 会議にてお知らせしています。

目　次

序　章　VZを使うまえに …… 0-1
0.1 VZエディタとは …… 0-2
0.2 Ver1.5の新機能 …… 0-4
0.3 動作環境 …… 0-5
0.4 マスターディスクの内容 …… 0-6
0.5 マニュアルの概要 …… 0-7
0.6 表記法について …… 0-8
0.7 謝辞 …… 0-10

第1章　VZエディタを使ってみよう …… 1-1
第2章　VZの起動 …… 2-1
2.1 コマンドラインの書式 …… 2-2
2.2 環境変数 …… 2-6
2.3 編集ファイル名の省略 …… 2-10
2.4 動作環境の設定 …… 2-12

第3章　ファイルの操作 …… 3-1
3.1 ファイル入出力 …… 3-2
3.2 マルチファイル …… 3-13
3.3 プロファイル機能 …… 3-15

第4章　スクリーンエディット …… 4-1
4.1 カーソルの移動 …… 4-2
4.2 基本編集コマンド …… 4-12
4.3 ヒストリー機能 …… 4-16
4.4 カット&ペースト …… 4-20
4.5 文字列の操作 …… 4-26
4.6 その他のコマンド …… 4-33
4.7 特殊文字の入力 …… 4-36

第5章　ファイラー …… 5-1
5.1 ファイラーの概要 …… 5-2
5.2 ファイラーの基本操作 …… 5-4
5.3 メニューコマンド …… 5-7
5.4 複数ファイルのコピー …… 5-16

第6章　DOSコマンドの実行 ……… 6-1
6.1 コマンドラインモード ……… 6-2
6.2 エイリアス ……… 6-6
6.3 ファイラーとの連係 ……… 6-9
6.4 コンソール出力の取り込み ……… 6-11

第7章　より高度な使い方 ……… 7-1
7.1 常駐モード ……… 7-2
7.2 EMS ……… 7-7
7.3 スワップ ……… 7-10

第8章　他機種版・付属ユーティリティ ……… 8-1
8.1 マルチ機種対応 ……… 8-2
8.2 J-3100版 ……… 8-3
8.3 IBM PC／AX版 ……… 8-6
8.4 EZKEY ……… 8-8
8.5 VMAP ……… 8-13

第9章　テクニカルノート ……… 9-1
9.1 モジュールの概要 ……… 9-2
9.2 アセンブル・リンク ……… 9-4
9.3 割り込みハンドル ……… 9-6
9.4 FEPの制御 ……… 9-7
9.5 グラフキーの利用 ……… 9-9

第10章 カスタマイズ ……… 10-1
10.1 カスタマイズの概要 ……… 10-2
10.2 DEFファイルの分割 ……… 10-4
10.3 DEFファイルの取り込み ……… 10-6
10.4 編集時の再カスタマイズ ……… 10-7
10.5 DEFファイルの書式 ……… 10-9
10.6 起動時オプション ……… 10-19

第11章　マクロ機能……11-1
11.1 マクロの概要……11-2
11.2 キーボードマクロ……11-4
11.3 外部マクロの実行……11-8
11.4 マクロのコーディング……11-11
11.5 マクロ言語仕様……11-18

付　録……12-1
A.コマンド一覧……12-2
B.オプション一覧……12-8
C.エラーメッセージ……12-12

索引……巻末

序　章 VZを使うまえに

0.1　VZエディタとは

0.2　Ver1.5の新機能

0.3　動作環境

0.4　マスターディスクの内容

0.5　マニュアルの概要

0.6　表記法について

0.7　謝辞

0.1 VZエディタとは

■ 高速・軽量

VZエディタは、MS-DOS標準形式のテキストファイルを編集するスクリーンエディタです。ですから、ワープロのような印刷機能や日本語入力機能はもっておりません。しかし、その分コンパクトで俊敏なレスポンスを備えています。パソコンに限らず、全てのツールにあるべく姿は、ユーザーがそれを意識しないで使えるということです。VZの設計思想の原点もそこにあります。

このところブックコンピュータと呼ばれるA4サイズパソコンの発表が各社から相次いでいます。これらは確かにセンセーショナルではありますが、やはり処理速度や記憶容量の点で、デスクトップタイプのパソコンには1歩も2歩もおよびません。しかし、そんな弱点を感じさせないのがVZです。VZは、まさにブックコンピュータ時代にふさわしい軽量ツールと言えるでしょう。

■ 操作性の追及

ただ高速・軽量であっても、使い難くては価値がありません。VZでは、こと細かな点に至るまで、操作性に気を配っています。いくつか例をあげてみましょう。

◎ ファイル名や検索文字列等の入力時には、常にヒストリー機能が働き、以前に入力した情報を有効に利用できます。

◎ 削除した文字・文字列または行・ブロックは、各々のバッファがあふれるまで順次記憶され、所定のコマンドで復活できます。

◎ 編集ファイルの指定時には、ディレクトリ、拡張子を指定する必要はありません。VZがあらかじめ設定したパスから、すばやくファイルを捜してきます。

■ DOS環境ツール

VZの上ではこのような操作環境が実現しました。でも、編集を終了しMS-DOSのコマンドラインに戻ったら、再びテンプレートの世界に逆戻りです。しかし御安心を。VZの環境を、そのままMS-DOSの世界に持ち込むことができます。それが「常駐機能」です。「AUTOEXEC.BAT」の中で「vz -z」と記述するだけで、コマンドラインでもVZの編集機能が使え、[**ESC**]キー一発で瞬時にVZが起動します。

みなさんは、ファイルのコピーや削除をどうやって行なっていますか？ ファイルの数が多くなると、いちいちタイプしていては大変ですから、専用ソフトを使われていることでしょう。実はVZもこの機能をもっています。ディレクトリを表示してファイルを選択し、指定コマンドを順次実行することもできるのです。

■ ユーザーフレンドリー

いくら操作性に気を配っても、やはり人の好みはさまざまです。VZは、決った操作を強要したりしません。キー割り当てはもちろん、メニュー、表示カラー、各種バッファサイズ等、ありとあらゆる機能を、DEF ファイルを書き変えるだけで自由に設定できます。また、馴染み易く強力なマクロ言語を装備しており、必要に応じて機能を拡張してゆけます。そしてパワーユーザーのためにソースコードも付属しました。使えば使うほど味が出てくる、そんなツールがVZです。

0.2 Ver1.5の新機能

VZエディタ Ver1.5で追加あるいは拡張された主な機能を以下に示します。

- ■ J-3100版、IBM　PC／AX版を標準添付しました。
- ■ EMSに対応しました。最大1MBまでのEMSメモリを、GVRAM用ワーク、テキストバッファ、テキストスタック、スワップバッファとして利用できます。
- ■ 高度なメモリスワップ機能を装備し、常駐サイズ2.5KB（-***SW***3指定時）を実現しました。また、必要に応じてVZの使用メモリを解放することができます。
- ■ テキスト管理ワーク、ラインバッファをコードセグメント外に確保することで、マクロ領域等を圧迫することなく、より多くのテキスト（-***TC***100以上）をオープン可能にしました。
- ■ ファイルを読むだけのViewモードを追加しました。テンポラリファイルを作成せず、より高速なファイルアクセスが可能です。
- ■ ファイルの検索機能を大幅に強化し、ディレクトリ・拡張子とも省略して、所定のパスからサーチ可能となりました。
- ■ バックアップディレクトリ機能を導入しました。「.bak」にリネームする代りに、環境変数「VZBAK」で指定したディレクトリへ、オリジナルファイルをセーブします。
- ■ ファイラーに幾つかのスクロールコマンドを追加し、より操作性を向上しました。
- ■ ファイル選択時の拡張子による指定コマンド実行機能を設けました。
- ■ ヒストリー機能にワード単位の検索機能を追加しました。
- ■ テキスト毎に表示カラム数を設定可能にしました。また、拡張子が数字の場合の、自動表示カラム数設定機能（-***Ea***指定時）を追加しました。
- ■ コマンドキー定義の重複制限を緩和し、[**ESC**][**X**]と[**Alt**]+[**X**]のような組み合わせを可能にしました。また、マクロも2つまでのキーを定義できます。
- ■ 拡張ファンクションキー（[**f･11**]～[**f･15**]）に対応しました。(-FK指定時)
- ■ 2バイト半角文字に対応しました。
- ■ メインメモリ、EMSの使用状況を表示する、「vmap.com」を添付しました。

※ Ver1.5の追加・変更機能の詳しい内容は「readme.doc」をご覧ください。

0.3 動作環境

VZの動作環境は、次のとおりです。

■ ハードウェア

【 PC-9801版 】

PC-9801シリーズ全機種（LTを除く）
PC-98XA／XL／XL2／RL
（ハイレゾリューションモード対応）
PC-286／386シリーズ

【 J-3100版 】

J-3100シリーズ日本語モード

【 IBM PC／AX版 】

IBM PCおよび互換機
J-3100シリーズ英語モード
AXマシン日本語モード

■ 動作OS

MS-DOS2.11以降（一部の初期バージョンを除く）

■ 最低必要フリーエリア

128KB（EMS非搭載時）
64KB（EMS搭載時）

0.4 マスターディスクの内容

VZエディタVer1.5のマスターディスクは、ルートディレクトリのファイルと、いくつかのサブディレクトリで構成されています。

VZ.COM		：VZ本体	(PC-9801版)
VZ.DEF		：標準DEFファイル	
VZ.DOC		：オンラインドキュメント	
VZS.COM		：スモール版VZ本体	(PC-9801版)
VZS.DEF		：スモール版VZ用DEFファイル	
EZKEY.COM		：EZKEY	(PC-9801ノーマルモード)
EZKEY.DOC		：EZKEY　ドキュメント	
VMAP.COM		：VMAP　メモリ情報表示ツール	
VMAP.DOC		：VMAP　ドキュメント	
START.BAT		：	
SAMPLE1.BAT		：　第1章　練習用ファイル	
SAMPLE2.BAT		：	
README.DOC		：最新版の情報、バージョンアップの内容	
J31	<DIR>	：J-3100版サブディレクトリ	
IBM	<DIR>	：IBM　PC／AX版サブディレクトリ	
SRC	<DIR>	：ソースコードサブディレクトリ	
*.DEF		：各種DEFファイル	

・最新版の正確な内容は「README.DOC」をご覧ください。
・VZを起動するには「VZ.COM」と「VZ.DEF」ファイルが必要です。
・「VZS.COM」はファイラー、エイリアス等の環境機能を除いた縮小版です。

0.5 マニュアルの概要

VZエディタVer1.5のマニュアルは、次に示す11の章から構成されています。このうちVer1.0の7,8章は10,11章に、そして7,8章は新たに書き加えられました。これは、特に参照頻度の高いマクロとカスタマイズの章を、付録とともに巻末にまとめたかったからです。

■　第1章 VZエディタを使ってみよう

エディタを使うのは初めて、という方は、まずこの章からお読みください。標準的なエディタの編集操作から初めて、VZエディタの簡単な使い方までを易しく説明します。

■　第2章 VZの起動

VZの起動の方法と、お使いの環境に合わせた環境変数、オプションの設定について説明します。起動時オプションの一覧は、第11章に移しました。

■　第3章　ファイルの操作

ファイルの入出力操作、および複数ファイルの扱いについて説明します。

■　第4章　スクリーンエディット

スクリーンエディタの種々のコマンドについて説明します。編集操作に関する説明は、この章までです。

■　第5章　ファイラー

ファイラーの活用方法について説明します。

■　第6章 DOSコマンドの実行

コマンドラインでの種々の拡張機能について説明します。MS-DOSのコマンドをあまりお使いにならない方は、読みとばしても構いません。

■　第7章　より高度な使い方

常駐モードの他、Ver1.5で拡張された、EMS、スワップ機能について説明します。

■　第8章　他機種版・付属ユーティリティ

他機種版の相違点、および付属ユーテイリティのEZKEY、VMAPについて説明します。

■ 第9章　テクニカルノート

VZをソースレベルで改良、または他機種への移植等をなさりたい方のために、モジュールの内容、アセンブル・リンクの方法、内部動作の留意点などついて説明します。

■ 第10章 VZのカスタマイズ

VZの種々の環境をカスタマイズする方法について説明します。起動時オプションもこの章でとりあげます。

■ 第11章 VZのマクロ機能

VZのマクロ言語を使ってマクロコードを作成するための言語仕様、およびコーディング、デバッグ方法について説明します。

■ 付録A コマンド一覧

VZ（PC-9801版）の全てのコマンドをまとめてあります。

■ 付録B オプション一覧

起動時オプションをまとめてあります。

■ 付録C エラーメッセージ

エラーメッセージと、その対処方法をまとめてあります。

0.6 表記法について

このマニュアルでは、次のような表記法を用いています。

【一般的な書式】

[...]　　括弧の項目を必要に応じて入力します（省略可）。
{ ... }　　括弧の項目を必要に応じて複数個入力します（省略可）。
< ... >　　該当する内容の入力を意味します。
<ｽﾍﾟｰｽ>　　一つ以上のスペースまたはタブを入力します。
<ﾀﾌﾞ>　　一つのタブコードを入力します。
... ...　　各項の間は<ｽﾍﾟｰｽ>で区切ります。
<n>　　数値（10進数）を入力します。
.. | ..　　いずれかの項目を入力します。
A>...　　DOSのコマンドライン入力を意味します。

【各種キーの表記】

^C　　[CTRL]+[C]
^QC　　[CTRL]+[Q][C]
[↵]　　リターンキー
[SPACE]　　スペースキー

【その他の表記】

TC10 DI-　　起動時オプション（英大／小文字は識別しません）
《··········》　　画面最下行に黄色で表示されるメッセージ
『··········』　　ウィンドウ、メニューのメッセージ
【··········】　　オプション

0.7 謝辞

■ PC-9801は、日本電気（株）の登録商標です。
■ J-3100は、（株）東芝の登録商標です。
■ IBM PCは、米国IBM社の登録商標です。
■ MS-DOSは米国MicroSoft社の登録商標です。
■ WordStarは米国WordStar社の登録商標です。
■ dBXLは米国WordTech社の登録商標です。
■ MASM,LINKは米国MicroSoft社の製品です。
■ TASM,TLINKは米国Borland社の製品です。
■ OPTASMは米国SLRsystems社の製品です。
■ Mifesはメガソフト社の製品です。
■ プリント専科IIは技術評論社の製品です。
■ LHarcはHaruyasu Yoshizaki氏の著作物です。

第1章 VZエディタを使ってみよう

本章は初心者の方のためのものです。ですから、ごく基本的な操作方法しか紹介しません。実はVZエディタにはもっともっと便利な機能があるのです。でも、本章の内容だけでも、十分実用になります。ワープロや歯切れの悪いエディタを使ったことのある方なら、VZエディタの使いよさがきっとお分かりになるでしょう。

なお、VZエディタでは、操作キーの設定は自由に変えられます。VZエディタを購入なさった状態でお使いになると、本章の説明では、WordStarを始めとして、世界の一流ソフトが使っている操作キーとほぼ同じになっています。

ディスクに組み込む

ＶＺエディタのマスターディスクには、PC-9801版、J-3100版、IBM PC/AX版の３種類のＶＺが入っています。まず以下の手順で、普段お使いのＭＳ－ＤＯＳシステムおよびカナ漢字変換の入ったフロッピィディスク、またはハードディスク（ハードＲＡＭ）へ、各機種版のＶＺをコピーして下さい。

・フロッピィディスクドライブの１番目へ、ＶＺのマスターディスクをセットします。誤ってディスクを書き換えないように、ライトプロテクト状態にしておいて下さい。
・フロッピィディスクへコピーしたい場合は、もう一方のドライブへセットします。
・サブディレクトリを作ってその中へコピーしたい場合は、予めサブディレクトリを作成し、そこへcd(change directory)しておきます。（*1）
・ＶＺのマスターディスクのドライブ名を打ち込みます。（*2）
・次のコマンドを打ち込みます。（*3）

```
A>install  ＜コピー先ドライブ名＞↵
```

・コピーが終了したら、「vz↵」でＶＺが起動します。（*4）

[例]（Ａ:からＣ:￥ＶＺへコピー）

```
(1)     C>md ¥vz↵
        C>cd ¥vz↵
(2)     C>a:↵
(3)     A>install c:↵
(4)     C>vz↵
```

※ ドライブＡからコピーする場合に限り、次の方法でもコピーできます。

```
(2)(3)  C>a:install↵
```

ここで、エディタの基本操作をご存じの方は、マスターディスクの「readme.doc」をお読み下さい。このまま導入編を読み進まれる方は、次の方法で、練習用のファイルもコピーしておいて下さい。

```
C>copy a:¥sample?.txt↵
```

コピーが終了したら、ＶＺのマスターディスクを抜いて、大切に保存して下さい。

VZエディタを起動する

使い方はごく簡単です。ただ

A>vz ↵

と打てば良いのです。すぐ下にタイトルが表示され、すぐに青色のウィンドウが出現します。ここに作成したり編集したりするファイル名を書けば良いのです。sample1.txtと打ってください。

```
A>vz
VZ Editor Version 1.50  Copyright (C) 1989 by c.mos

┌【入力ファイル】 <A:¥> ──────────────┐
│sample1.txt                        │
└───────────────────────────────────┘

  C1  CU  CA  S1  SU    VOID  NWL  INS  REP  ^Z
```

リターンキーを押すと、瞬時に画面が変わり、

```
 P □      1:1                               1  SAMPLE1.TXT
↓
*  本ファイルは、VZエディタの入門用の練習ファイルです。↓
↓
 1. The quick brown foxes jumped over the lazy dog.↓
 2. The quick brown foxes jumped over the lazy dog.↓
 3. The quick brown foxes jumped over the lazy dog.↓
 4. The quick brown foxes jumped over the lazy dog.↓
 5. The quick brown foxes jumped over the lazy dog.↓
 6. The quick brown foxes jumped over the lazy dog.↓
 7. The quick brown foxes jumped over the lazy dog.↓
 8. The quick brown foxes jumped over the lazy dog.↓
 9. The quick brown foxes jumped over the lazy dog.↓
10. The quick brown foxes jumped over the lazy dog.↓
11. The quick brown foxes jumped over the lazy dog.↓
12. The quick brown foxes jumped over the lazy dog.↓
↓
cover discover recover↓
covered covers↓
↓
*  VZは、次のような数々の特徴を備えた、高性能ツールです。↓
↓
↓
*【高性能エディタとして】↓
  ファイル  窓切換  文切換  窓分割  記憶    検索  置換  カット  インサート  ブロック
```

のようになります。指定したファイル名が既に存在するものである場合には、そのファイルの内容が画面に現れます。

なお指定したファイルが見つからない場合は、

《"sample1.txt"は見つかりません. 新規ファイルですか? (Y/N)》

と聞いてきます。ここで[Y]を押すと、sample1.txtという新しいファイルを作成することになります。新しく文書を作る際には、この方法を使ってください。

■ ステータス・ライン

画面の一番上の一行は、ステータス・ラインといって、現在の編集状況を表します。左から順に説明しておきましょう。

ページング・モード

一番左の英字は、ページ・スクロールの際のモードを表しています。後でくわしく説明しますが、通常は［P］になっていなければいけません。そうでない場合には、^@を打って［P］に戻しておいてください。^@とは、キーボードの左端にあるコントロールキー［CTRL］を押しながら、［@］を打つことを表しています。^@ を打つと、［P］→［C］→［P］と文字がくるくる変わります。

ブロック・モード

通常は白い四角ですが、ブロック・モードになると黒い四角になります。これについては後で説明します。

カーソル・ポジション

［1:1］のように、［:］で区切った2つの数字があります。これは［行:桁］を表しています。［1:1］は、現在のカーソルの位置が1行目の1文字目にあることを表します。

インジケータ

黒い長方形があります。これは、現在のカーソルがファイル全体のどこら辺に位置しているかを示しています。

テキスト・ナンバ

VZは、複数のファイルを一度に読み込むことができます。この数字は、読み込んだ順序を表す番号です。

修正モード

ここに［*］を表示しているときには、現在見ているファイルに修正を行ったことを表しています。また、［R］を表示している場合は、書き込み禁止であることを示しています。「書き込み禁止」については後で説明します。

ファイル名

現在編集を行っているファイルの名前です。

VZエディタを終了する

VZを終了するには、キーボードの左上にあるエスケープキー[ESC]を押してから[Q]を押します。新しいテキストを書いたときや、テキストを修正したときは、画面の一番下に、

《修正テキストを出力しますか？ (Y/N)》

という表示が出ます。ここで[Y]を押すと、現在編集中のテキストをディスクに書き込みます。また[N]を押すと、書き込みを中止します。この場合は、修正または制作したテキストはなくなってしまいますからご注意ください。今は書き込みも修正もしていませんから、[N]を押してください。

次に、

《エディタを終了しますか？ (Y/N)》

という表示が出ます。[Y]を押すとエディタを終了してDOSに戻ります。ここで[N]を押すと、さらに編集を続けることができます。

カーソルの移動（1）

もう一度VZエディタを起動してください。今回は、編集するファイル名も打ってしまいましょう。このように、ファイル名を一緒に打つこともできるのです。

A>vz sample1.txt↵

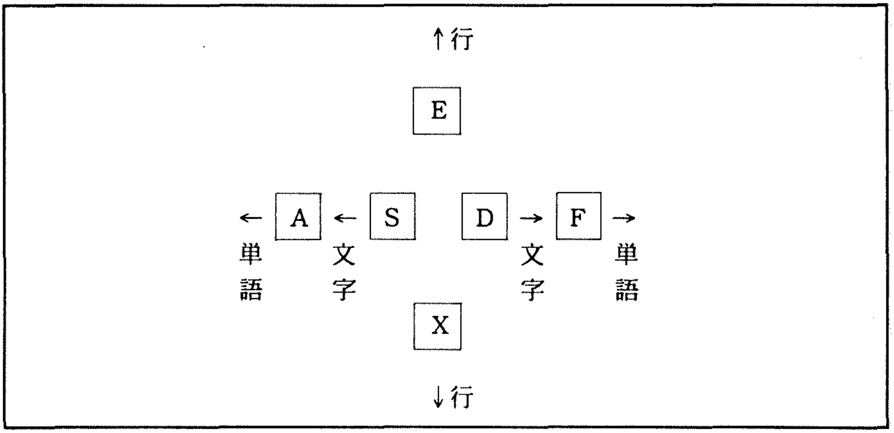
先ほどと同じ画面になりましたね。

点滅している白い四角をカーソルといいます。カーソルは「この位置に文字を入力できますよ」という印ですから非常に重要です。そこでまず、カーソルを動かす練習をして見ましょう。VZエディタを購入したままお使いの場合は、世界のパソコンのソフトで圧倒的に取り入れられているマイクロプロ社（以下ワードスタータイプと呼びます）の方式に似せてあります（VZエディタの操作方法は、自由に変えることができます。これにつきましては、第10章をご参照ください）。

操作の基本は、コントロールキー（キーボードの左の方にある、[CTRL]と書いてあるキーです）と英字キーを同時に押して行います。今後 ^F と記述してある場合には、コントロールキーを左の小指で押しながら、[F]を押してください。なお、矢印キーなどの特殊キーでも使うことができますので、特殊キーに慣れている方はそちらをお使いください（特殊キーについては、括弧の中に記述してあります)。

ワードスタータイプの操作キーは、カーソルの移動については、下の図のように、視覚的に憶えやすい位置になっています。多くのソフトで採用されており、過去のタイピスト達が愛したものですから、この操作に熟練すると非常に便利です。

なお漢字モードになっている場合でも、仮名シフトキーを押して仮名モードになっている場合にも、これらのコマンドは英字半角モードと同じように使うことができます。



まずカーソルを上下左右に動かしてみましょう。1文字または1行カーソルを動かすには、

1行上へ ……… ^E ,[⬆]
1行下へ ……… ^X ,[⬇]
1文字左へ …… ^S ,[⬅]
1文字右へ …… ^D ,[➡]

を使います。カーソルが動くと、画面一番上のカーソル・ポジションの数字が変わるでしょう。画面のあちこちにある青い［↓］は改行記号です。この改行記号を越えてカーソルを動かすと、カーソルは次の行に移ります。

また、画面の外へカーソルを動かそうとすると、画面がスクロールします。^X を押し

続けてみましょう。カーソルが一番下から２行目のところへいくと、もうカーソルが動かず、テキストが上にスクロールしていきます。相対的に見れば、カーソルが下へ動いているわけです。本マニュアルでは、これをスクロールダウンと呼びます。

この反対に ^E を押し続けると、カーソルが画面の一番上に入ってからは、テキストの方が下に流れてきます。

お分かりになったでしょうか。カーソルをいろいろ動かして、サンプルの文章の中を歩きまわってください。

次に、単語ごとのカーソル移動を実験してみましょう。

１単語左へ …… ^F ，［SHIFT］+［➡］
１単語右へ …… ^A ，［SHIFT］+［⬅］

矢印キーを使うときは、［SHIFT］を押しながら矢印キーを打ってください。

カーソルを英文の書いてある行の一番左にもってきてください。この英文の意味などは気にしないでください。ただアルファベットの全ての文字を含む文章というだけのことです。かつて、英文タイプを購入するときのテストによく用いられた文章なのです（あえて意味を書くと、「機敏な狐達が、怠け者の犬を飛び越えました。」ということになります）。

1. The quick brown foxes jumped over the lazy dog.

^F を押してみましょう。1, T, q, b, fと、単語１つずつカーソルが右に移動するでしょう。英文を書くときや、プログラムを書いているときには便利ですね。同じように、^A を押してみましょう。今度は単語ごとにカーソルが左に移動します。

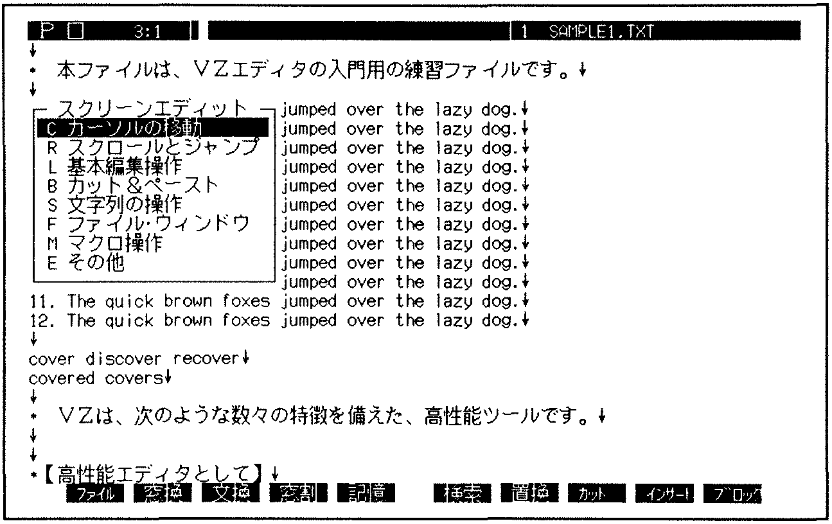
では日本語のときはどうでしょうか。カーソルを一番上の行の「本」のところにもっていってください。^F を打つと、「V」、「入」、「練」と、漢字や全角英字のところへ飛んでくれますね。

この単語の区切りに関する細かい規則は、第４章の「単語の区切り」のところに書いてありますから、参照してください。

でも操作方法を忘れたときは困りますね。困ったときは助けて欲しいでしょう。そうです、［HELP］というキーを押せばよいのです。［HELP］キーは文字どおりお助けキーで、キーボードの右のテンキーの上にあります。［HELP］を押すと、画面に小さなウィンドウ

が開いて、ヘルプのメニューが出ます。

メニューから必要な項目を選択するには2つの方法があります。



一つはカーソル移動キー（今勉強しました）を使って、必要な項目を反転し、リターンキーを押します。もう一つは、項目の前に書いてある英字をポンと打つのです（この場合は、かなロックをしてあるとうまくいきません)。慣れてくると、この方が速いでしょう。今わからなくなったのはカーソルの移動ですから、[C] を選びます。画面に

のように、「カーソル移動」に関するコマンドキーと簡単な説明が表示されます。

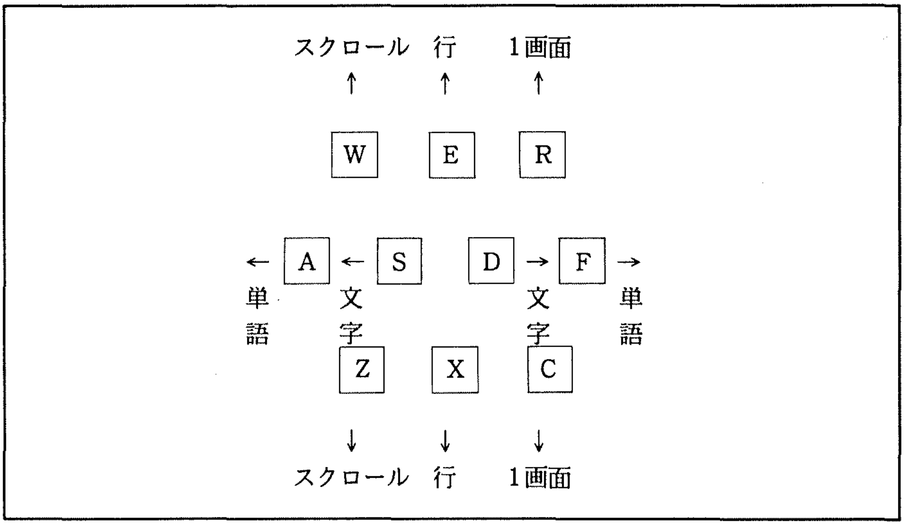
[#05] や [e] という記号もいっしょに表示されますが、これは第8章で述べるマクロをプログラムする時便利な情報ですから、関心の無い方は無視してください。

ここでリターンキーを打てば再びヘルプメニューになります。もうヘルプは必要ないという場合は、[ESC]を打てば編集に戻れます。なお、VZエディタでは、以前の状態に戻る際には[ESC]を打つことになっています。わからないときには[HELP]、困ったときには[ESC]、と憶えてください。

VZエディタには沢山の機能がありますので、思わぬウィンドウが開いたり、わけの分からないメッセージが出たりすることがあるかも知れません。画面の一番下に何かの指示が出た場合以外は、[ESC]を押すと元の状態に戻ります。

カーソルの移動（2）

文章を書いているときには、もっと大幅にカーソルを移動したいときがありますね。例えば、画面のすぐ上の行を見たいとか、すぐ下の行を見たいときなどです。



この場合は、

1行スクロール・アップ ……　^W ，［SHIFT］+［ROLL　DOWN］
1行スクロール・ダウン ……　^Z ，［SHIFT］+［ROLL　UP］

を使います。^W を押すと、テキストが下に1行ずつ流れて、上の1行を見ることができます。その際に、カーソルの位置は変わりませんね。^Z は、上下の動きが反対なだけです。

ここで、「スクロール・アップ」の「アップ」は、カーソルがテキストの上（前）の方向に移動するため、「アップ」と呼ぶことに注意してください。[ROLL UP]、[ROLL DOWN]キーの名称は、「画面が動く方向」を表していますので、これとは反対の考え方になっています。

次に1画面ずつカーソルを動かしてみましょう。

1画面上へ …… ^R , [SHIFT]+[↑]
1画面下へ …… ^C , [SHIFT]+[↓]

^C を押すと、22行ずつ画面が上に流れます。ここで ^C を押し続けてみましょう。あっという間にテキストの終りになりましたね。VZエディタの速さにビックリなさったことでしょう。なお一番最後にある青い [←] は、ファイルの最後の印です。^R を押してテキストの一番最初に戻ってください。

^R や ^C がうまく動かないことがときどきあります。それは、このキーにいろいろな機能が隠されているからです。画面の一番上の左端の表示が [P] になっていることを確認してください。もし [S] や [C] になっていたら、^@ を打って、[P] に変えてください。この点については後で説明します。

なお本節のヘルプは、ヘルプメニューの『R スクロールとジャンプ』の中にあります。

文章を書く

カーソルを動かすだけではつまりませんから、何か文章を書いてみましょう。カーソルを12番目の英文の下の行の最初に持っていってください。13番目の英文として、上の行を書き写してみましょう。

12. The quick brown foxes jumped over the lazy dog.
13. The quick brown foxes jumped over the lazy dog.

最後の改行は、リターンキーを押してもよいですし、^M でも同じ働きをします。ここで、^S を2回押して、カーソルを [dog] の後の [.] の上に持ってきてから [s] を押してください。[dog] が [dogs] に変わり、犬が複数になりましたね。既に文字が書いてある上に別の文字を打ち込むと、打ち込んだ文字が前の文字の前に入ります。これを

「挿入モード」といいます。VZエディタでは、カーソルが白い四角になっているときは、挿入モードです。

挿入モードの切替え …… ^V，［INS］

ここで ^V を押してください。カーソルがアンダーラインになりましたね。これは「上書きモード」に入ったことを表しています。そこでカーソルを [fox] の [f] の上において、[cat] と打ってください。「狐」が「猫」に変わったでしょう。「上書きモード」では、打ち込んだ文字が下の文字と交換になるのです。
^V は、一度押すたびに、「挿入モード」から「上書きモード」へ、さらにまた「挿入モード」へと、くるくる変わります。[INS]でもできます。

日本語を打ち込むときも全く同じです。お使いのかな漢字変換を使って、同じように打ち込んでください。

本ファイルは、VZエディタの入門用の練習ファイルです。

の後に、ご自分の名前でも打ち込んでください。

何ですか。間違った文字を打ち込んでしまったのですか。では、1文字削除のキーをお教えしましょう。

バックスペース …… ^H，［BS］
1文字削除 ……… ^G，［DEL］

消したい文字の上にカーソルがあるときは、^G を打ってください。もちろん[DEL]でも結構です。間違った文字を打ってすぐ気がついたときは、カーソルは消したい文字のすぐ右にありますから、カーソルの左の文字を消せると便利ですね。その場合は、^H を打ってください。[BS]でも同じ働きをします。では削除の練習をしてみましょう。カーソルを最後の英文の [quick] の後ろのスペースのところに置いてください。

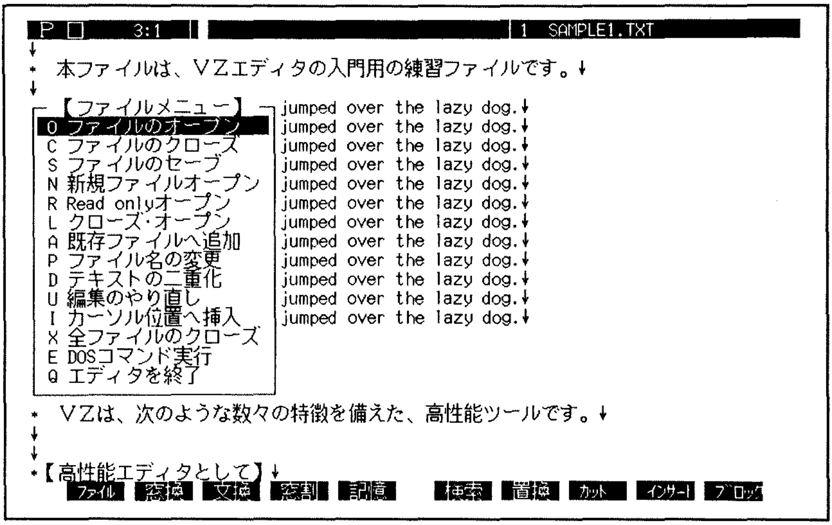
12. The quick brown foxes jumped over the lazy dog.

ここで ^H を5回打ちます。「狐」が機敏 (quick) でなくなりましたね。このような削除機能を「バックスペース」といいます。

次に、[brown] の [b] の上にカーソルを移してください。ここで ^G を5回打ちます。

今度は「狐」の色がなくなりましたね。^G はカーソル位置の文字を消すのです。

そろそろ疲れてきたので、ファイルをしまっておきましょう。今回は、[**ESC**][**Q**] を打つ代わりに[**f･1**]を押してみましょう。画面にウィンドウが現れました。



いろいろなメニューがあります。くわしくは後で説明しますが、さしあたって [Q] でよいのでしたね。カーソルを『Q エディタを終了』のところへ持っていってリターンキーを打ってもよいですし、ただ[**Q**]と打ってもよいのです。

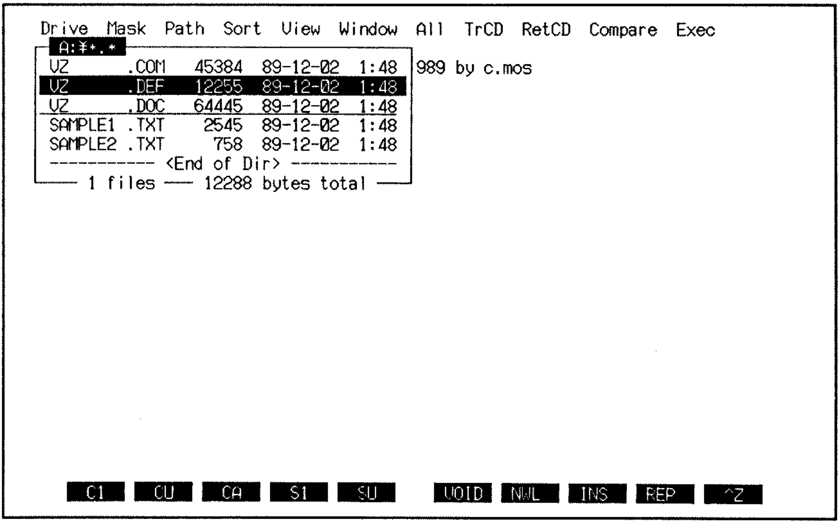
エディタを終了して、MS－DOSのプロンプトが出たら、お茶にしましょう。

カーソルの移動（3）

一休みしたら、また勉強を始めましょう。今度は一番楽な起動方法をやってみましょう。ただ

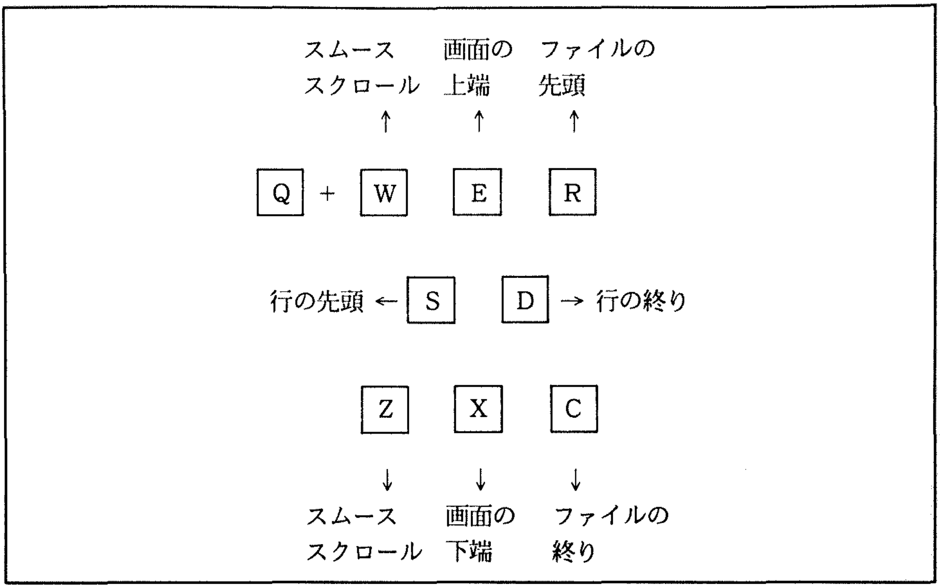
A>vz↵

と打って、入力ファイルを聞いてきたらリターンキーを打ってください。画面にファイル名の一覧が表示されます。



VZエディタでは、この画面をファイラー画面と呼びます。ここではいろいろなことができますが、詳しくは第5章をご覧ください。

今回は、ただオープンするファイルを選ぶだけにします。カーソルを動かして［sample1. txt］のところに持っていき、リターンキーを押してください。これで［sample1. txt］を読み込みました。



VZエディタでは、2つのキーを続けて押す操作方法があります。コントロールキーを押したまま、[Q], [D]と続けて2つのキーを押してください（厳密にいえば、[D]を打つ時にはコントロールキーを離してもよいのですが)。カーソルが行の終りに行きましたね。これを^QDと書くことにします。

行の先頭へ ………… ^QS ，[CTRL]+[←]
行の終りへ ………… ^QD ，[CTRL]+[→]
画面の上端へ ……… ^QE ，[CTRL]+[↑]
画面の下端へ ……… ^QX ，[CTRL]+[↓]
ファイルの先頭へ …… ^QR ，[CTRL]+[ROOL DOWN]
ファイルの終りへ …… ^QC ，[CTRL]+[ROLL UP]

^QD と ^QS は反対の動きをします。^D や ^S は、1文字ずつ左右に動きますが、その前に ^Q を付けると、行毎に動くのです。ここで行というのは、改行マークの次の文字から、次の改行マークまでを指します。画面上の「1行」とは違いますのでご注意ください。

なお、画面表示の行の最初や最後に移動するときには、^Q[と ^Q] を使います。いろいろ動かして、この2種類のキーの動きの違いを実験してみてください。

^QE と ^QX は、画面の上端または下端の1行上へカーソルを移動します。このように、^QS, ^QD, ^QE, ^QX または [CTRL]+[矢印キー]を使えば、カーソルを画面の上下

左右の端にもっていけるわけです。

次は ^QR と ^QC です。^R や ^C は、1画面上下に動かすキーでした。前に ^Q が付くと、ファイルの最初や最後に移動する機能に変わります。^QR と ^QC を交互に打って、VZエディタのスピードを楽しんでください。何万字もある大きなファイルでも、一瞬にジャンプすることができますよ。超高速がVZエディタの特徴なのです。

ここで、ちょっと難しい話をさせてください。いかに高機能なVZエディタでも、数百KBという特大のファイルを処理する場合には、一時的記録のためにディスクを利用します。ですからディスクに空き領域がないと困ってしまうのです。もし、非常に大きいファイルを編集する際には、一度 ^QC でファイルの終りに移動してみてください。これで何も言われなかったら、編集を始めても大丈夫です。

最後は、VZの売り物の一つ、スムース・スクロールです。

スムース・スクロール・アップ …… ^QW ,[ROLL DOWN]
スムース・スクロール・ダウン …… ^QZ , [ROLL UP]

[CTRL]を押しながら、[Q]を押して、さらに[Z]を押したままで画面を見てください。テキストがゆっくり下に流れていきますね。画面のちらつきもなく、とても見やすいはずです。指を[Z]のキーから離すと画面が止まります。編集している文書全体を眺めたいときに便利ですね。^QW では反対に動きます。

おもしろいことに、このキーは打ち方によって違った動きをします。今度は、[CTRL]を押したまま、[Q]を打って、さらに[Z]をポンと軽く打ってください。今度は、キーからすぐ指を離すのです。画面は同じように流れていきますが、止まりません。ここで数字の[0]から[4]を押してみてください。数字が小さくなるにしたがって、流れるスピードが遅くなりますね。そうです。自分の好きな速さに調節できるのです。

どうやって止めるのか、ですって。それは簡単です。[0]から[4]の数字以外のキーをどれか押せばよいのです。

また、スムース・スクロールしている時に[SHIFT]を押すと、押している間、スクロールのスピードが速くなります。

この機能は、2つのファイルを処理しているときには特別の機能を持っています。これについては後で説明します。

いろいろな削除

基本的なカーソルの移動はこれでおしまいです。いよいよテキストの編集の練習に入ります。

まず最初は削除の仕方です。1文字削除（^G，^H または［DEL］，［BS］）については前に勉強しましたね。では間違えて必要なものを消してしまったらどうすればよいのでしょうか。カーソルを

12. The quick brown foxes jumped over the lazy dog.

の［dog］のところへ持っていって、^G を3回打ってください。犬がいなくなってしまいましたね。でもあわててペット屋さんに行く必要はありません。^U を3回押してください。VZエディタは、消した文字をちゃんと憶えていて、元へ戻してくれるのです。11番の英文の［dog］辺りから初めて、^G を押したままにして次の行の中程まで消してみてください。ここで ^U を押し続ければ、今消した文字が皆出てきます。

誤って文字を消してしまったら ^U と憶えておいてください。［Undo］の［U］ですから憶えやすいですね。

次に1行の削除をやってみましょう。11番の英文の上にカーソルを置いて、^Y を打ってください。1行全部が消えましたね。消した行は復活できるでしょうか。これもやってみましょう。1番の英文の上にカーソルを置いて ^Y を押し続けて、英文を全部消してください。大変なことになりましたね。ここで ^J を押してください。消えた行が生き返ったでしょう。ずっと押しつづければ全ての行が生き返って元に戻ります。

文字の削除には、他にもいろいろな方法があります。説明は省略しますが、一覧を載せておきますから、いろいろ実験してみてください。間違えたら ^U で生き返りますから、安心して実験ができるはずです。

バックスペース ………… ^H，［BS］
1文字削除 ……………… ^G，［DEL］
単語の先頭まで削除 …… ^QH，［SHIFT］+［BS］
単語の終りまで削除 …… ^T，［SHIFT］+［DEL］
行の先頭まで削除 ……… ^QT，［CTRL］+［BS］
行の終りまで削除 ……… ^QY，［CTRL］+［DEL］
削除した文字の復活 …… ^U
1行削除 ………………… ^Y，［f・8］
削除した行の復活 ……… ^J，［f・9］

検索

文章やプログラムを編集していると、特定の言葉がどこにあるか探したい場合があります。このような場合には ^QF を押してください。

検索文字列の設定 …… ^QF，［f・6］

画面に、

```
┌【検索文字列】──────────────┐
│                                  │
└──────────────────────┘
```

というウィンドウが開きます。ここで

foxes↵

と入力してください。画面の左下に黄色く［foxes］という表示が出ます。さらに、ステータス・ラインの左上にあった［P］が黒い［S］に変わります。これは現在検索モードになっていることを表わしています。

ここで、^C を押してください。現在のカーソルより後ろにある最初の［foxes］のところにカーソルが移動します。［foxes］を探し出したわけです。続けて ^C を押せば、ファイルの終りに向かって次々と［foxes］を探し出します。もう［foxes］がなくなった場合には、画面の一番下に黄色く

《文字列が見つかりません》

という表示が出てカーソルは移動しません。それでは、^R を打ってみてください。カーソル位置からファイルの最初に向かって［foxes］を探し始めます。このように、^R と ^C で、カーソル位置から前後に検索することができます。

検索が終ったら、^@ を打って、ステータス・ラインの［S］を［P］（通常モード）に戻しておいてください。そうしないと、^C や ^R が1画面スクロールになりません。

検索する文字列は、完全に一致するものだけです。「一致する」かどうかは、単語の区切りと関係があります。スペースや記号を単語の区切りと解釈しますので、英語などの半角文字の場合は、スペースや記号で区切られた単語と同じことになります。［over］を探しているときは、［cover］は一致しない文字列と考えますし、［he］を探しているときは、［then］は違う文字列と考えます。日本語の場合は、英単語のように明確な区切りがありませんから、前後の文字に関わらず、同じ文字列があったら一致したとみなします。

ある文字列を含む単語を見つけ出したいときは、[*] を前後に付けます。例えば、

```
*over
```

を指定すれば、

```
cover discover recover
```

を探してくれます。また

```
cover*
```

を指定すれば、

```
covered covers
```

を見つけてくれます。ですから、

```
*cover*
```

のように [*] を前後に付ければ、[cover] という文字の並びを含む単語を全て探してくれます。

この他、改行マーク付きの文字列、例えば「ください。↓」などを探したいときもあります。この場合は、検索文字列として

```
ください。¥n
```

と指定してください。[¥n] を改行マークと解釈します。

置換

文章を編集している場合に、一定の言葉を違う言葉と取り替えたい場合があります。例えば、山田さん宛の手紙を渡辺さん宛てに変えるときなど、文中の「山田」をすべて「渡辺」に変えることができれば便利ですね。これを「置換」といいます。

文字列の置換 …… ^QA ,［f･7］

まず ^QA と打ってください。

```
┌【検索文字列】─────────────────┐
│                              │
└──────────────────────────────┘
```

というウィンドウが開きますから、

dog↵

と入力してください。次に、

```
┌【置換文字列】─────────────────┐
│                              │
└──────────────────────────────┘
```

というウィンドウが開きます。これに対しては

cat↵

と入力します。今度は

```
┌【置換範囲】──────┐
│ A テキスト全体   │
│ F カーソル以降   │
│ B カーソル以前   │
└─────────────────┘
```

というウィンドウが開きます。これは、置換を行う範囲を聞いているのです。ここでは[A] を選んでファイル全体で一度に置換をしてみましょう。[A]を打つと、画面の一番下に黄色い文字で、

《全部一度に置換しますか？ (Y/N)》

と聞いてきます。[N]を打つと、置換すべき文字列を見つけるたびに、

《置換しますか？ (Y/N)》

と聞いてきて、一つ一つ確認しながら置換を行うことができます。ここでは、[Y]を打って、一度に置換をしてみましょう。あっというまに13匹の犬が猫に変わったでしょう。VZエディタの置換の速さにびっくりなさったことと思います。カーソルもちゃんと以前の位置に戻っているでしょう。

では、もう一度猫を犬に戻しましょう。今度は怠慢なやり方をお教えします。『検索文字列』を聞いてきたときに、^E ([⬆]) を打ってみてください。今まで文字列を要求されたときに打ち込んだものが次々と出てきます。^E と ^X ([⬆]と[⬇]) を使って [cat] を探し出してリターンキーを打ってください。「置換文字列」に対しても同じように [dog] を探し出してください。これで1文字も入力すること無しに仕事をすることができました。VZエディタは、以前に入力した文字列をちゃんと憶えているのです。

文字列やファイル名を入力するウィンドウでは、以前に入力したものを呼び出すことができるだけでなく、カーソルを動かしたり、文字を上書きしたり挿入したり、また文字を削除するなどの編集もできます。以前に習ったコマンドをいろいろ試してみてください。

先ほど検索のところで、^@ を打っていろいろモードを変えましたね。画面の左上の表示で、

[P] ：普通のスクロール
[S] ：検索モード

でしたね。では [C] は何でしょう。これはタイトルのある行を探し歩くモードです。タイトル行とは、行の最初の文字が、

* / ; ^L (^L は改ページマーク)

で始まり、その上の行が空いている行（改行マークのみ）を指します。

では、^QRを打ってカーソルを文書の先頭に戻してから ^@ を押して、タイトルサーチ・モード [C] にしてください。そこで次々と ^Cを押すと、

* 本ファイルは、VZエディタの入門用の練習ファイルです。
* VZは、次のような数々の特徴を備えた、高性能ツールです。
* 【高性能エディタとして】
* 【DOS環境ツールとして】
* 【ユーザーフレンドリーな設計】

の5つの場所でカーソルが止まります。タイトルの部分に * などを付けておけば、タイトルだけを選んでスクロールすることができるのです。

文章の切り張り

コンピュータで文書を扱っていて一番うれしいのは、文の一部をあちこち切ったり貼ったりできることです。今回はこの機能を使ってみましょう。

カーソルを英文の一番上に持っていって ^B を打ってみましょう。画面上部の白い四角（□）が黒い四角（■）に変わったことにご注意ください。これは「ブロックモード」になったことを表しています。次に、^X を打ってカーソルを下にさげて、最後の英文のところまで持っていってください。英文が全部青色になりましたね。これで、英文の部分をすべてブロックとして指定しました。

```
P □   11:1                                   1  SAMPLE1.TXT
↓
*  本ファイルは、ＶＺエディタの入門用の練習ファイルです。↓
↓
 1. The quick brown foxes jumped over the lazy dog.↓
 2. The quick brown foxes jumped over the lazy dog.↓
 3. The quick brown foxes jumped over the lazy dog.↓
 4. The quick brown foxes jumped over the lazy dog.↓
 5. The quick brown foxes jumped over the lazy dog.↓
 6. The quick brown foxes jumped over the lazy dog.↓
 7. The quick brown foxes jumped over the lazy dog.↓
 8. The quick brown foxes jumped over the lazy dog.↓
 9. The quick brown foxes jumped over the lazy dog.↓
10. The quick brown foxes jumped over the lazy dog.↓
11. The quick brown foxes jumped over the lazy dog.↓
12. The quick brown foxes jumped over the lazy dog.↓
↓
cover discover recover↓
covered covers↓
↓
*  ＶＺは、次のような数々の特徴を備えた、高性能ツールです。↓
↓
↓
*【高性能エディタとして】↓
  ファイル 窓換 文換 窓割 記憶   検索 置換 カット インサート ブロック
```

ここで ^KK を打ってみましょう。黒い四角は白に戻り、青色は消えました。VZエディタは、今のブロックをしっかり記憶したのです。では ^QC でカーソルをファイルの終りに持っていきましょう。そこで ^J を打ってください。先ほどの英文がファイルの一番下に移りましたね。このようにして、文章の一部を好きなところに移すことができます。

今説明した ^J では、文章を移した瞬間にVZエディタは記憶を失ってしまいます。では、記憶したブロックを複数箇所に移すにはどうすれば良いでしょうか。次にこの操作をやってみましょう。

1番の英文のところで ^B、それから ^X を2度打って3番の英文へカーソルを移動し、^KK を打ってください。これで1番と2番の英文を記憶しました。次に、^QC でファイルの終りに行って、^KC を3回打ってみてください。先ほどの2行が3つ現れたでしょう。このように ^KC はブロックを複写しても記憶はなくならないのです。

ブロックモードの切替え …… ^B，［f･10］
行･ブロックの削除 ………… ^Y，［f･8］
行･ブロックの挿入 ………… ^J，［f･9］
行･ブロックの記憶 ………… ^KK，［SHIFT］+［f･8］
行･ブロックの複写 ………… ^KC，［SHIFT］+［f･9］

ブロック指定を利用すると、いろいろなことができます。まずブロック削除をやってみましょう。英文の部分全体をブロック指定（^B ^X ^X ^X）して ^Y を押してみてください。アッ大変、英文がすべて消えてしまいました。VZエディタは超高速ですが、沢山の行を削除するのに ^Y をベタ打ちするのは大変です。ブロック削除を使った方が楽ですね。

ところで、英文がなくなってしまっては、これからの勉強が面倒です。^J を打ってください。狐と犬の大群はすべて生き返りました。VZエディタのスピードと使いやすさを満喫されたことと思います。

実は、^KK と ^Y の違いは、画面のブロック部分を消すか消さないかの違いだけで、どちらも同じところに記憶しておくのです。

この機能を利用すると、文章の一部を別の場所に移動することができます。まず1番から3番の英文をブロック削除して、カーソルを12番の英文の下に置きます。 ここで ^J を打つと、3つの英文が見事に移動しました。1行だけ移動するのなら、^Y と ^J だけでできますね。ブロック関係と行削除も記憶しておくところは同じなのです。

VZエディタのブロック指定は、画面の一部を四角に取り込むことはできません。でも、一部を取り込むことはできます。カーソルを英文の［The］の上に置いて ^B を打ってください。それから、［foxes］の後ろにカーソルを移動して、［The quick brown foxes］をブロック指定してください。

1. The quick brown foxes jumped over the lazy dog.

^KK を打って、適当なところにカーソルを持っていって ^J を打ちます。「機敏な茶色の狐くん」が、さっと現れましたね。

このように、行の一部をブロック指定することができるのですが、ブロック指定が複数の行にまたがったらどうでしょうか。^B を打ってから ^D または ^F で、カーソルをどんどん右へ進めて、次のようにブロック指定をしてください。

1. The quick brown foxes jumped over the lazy dog.
2. The quick brown foxes jumped over the lazy dog.
3. The quick brown foxes jumped over the lazy dog.

同じように ^KK でブロックを記憶し、^J で適当なところに挿入してみてください。

the lazy dog.
2. The quick brown foxes jumped over the lazy dog.
3. The quick brown foxes

カーソル位置に記憶したブロックの内容が現れました。このように、ブロックモードでは、カーソルを上下に動かしている間は行単位でブロックを指定し、カーソルを左右に動かすと、文字単位でブロックを指定します。

ブロック内の置換

ブロックを利用すると、特定の範囲にある文字列だけを置換することができます。1番から3番までの英文をブロック指定して、

1. The quick brown foxes jumped over the lazy dog.
2. The quick brown foxes jumped over the lazy dog.
3. The quick brown foxes jumped over the lazy dog.
4. The quick brown foxes jumped over the lazy dog.
5. The quick brown foxes jumped over the lazy dog.

ここで、茶色の狐を赤くしてみましょう。「置換」のやり方を憶えていますか。
まず ^QA を押してください。そして、次のように入力または選択してください。

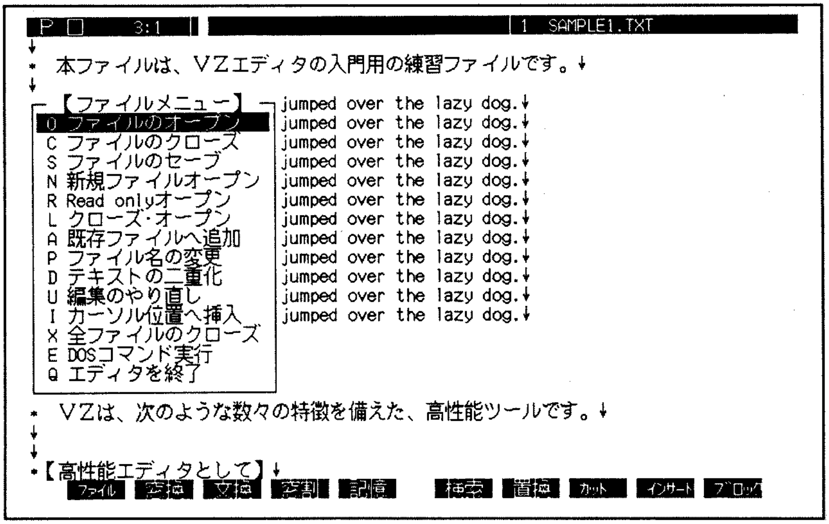
『検索文字列』 ・・・・・・・・・・・・・・ brown
『置換文字列』 ・・・・・・・・・・・・・・ red
『置換範囲』 ・・・・・・・・・・・・・・・・ [A テキスト全体]
《全部一度に置換しますか》 ・・ [Y]

1番から3番の狐だけが赤くなりましたね。ブロックの青い反転が目障りですので、^B を打って消しておきましょう。このように、ブロックと置換を組み合せると、好みの部分だけで置換を行うことができます。

この機能を利用して、特定の部分の特定の文字列を消すこともできます。消す際には『置換文字列』のところで、ただリターンキーを打てばよいのです。

複数のファイルを開く

[f･1]を押すと、次のような『ファイルメニュー』というウィンドウが開きます。



これは、ファイルの操作に関するコマンドの一覧で、カーソルを使いたいコマンドの上に動かしてリターンキーを押せば実行できます。また先頭の英字1字を打ってもよいのです。この方がスピーディかも知れません。ウィンドウの一番下を見てください。「Q エディタを終了」というのがありますね。以前に終了の際は[ESC]を押してから[Q]を押すと説明しました。実は『ファイルメニュー』にあるコマンドはすべて [ESC]+[英字1字] で実行することもできるのです。このように、憶えてしまったコマンドは、[ESC]+[英字1字] で実行してください。この方が簡単ですから。

今後は、[ESC]を押してから[O]を押すことを[ESC]Oと書きます。

■ O ファイルのオープン ([ESC]O)

現在 [sample.txt] を編集中ですね。ここで新しいファイルをオープンするとどうなるのでしょうか。『ファイルメニュー』で [O] (Open) を選択して新しいファイル [sample2.txt] をオープンしてみましょう。画面ががらりと変わって [sample2.txt] が現れました。

実は、これがVZエディタのマルチファイル機能なのです。現在［sample1.txt］と［sample2.txt］の2つのファイルを処理しているのです。えっ何ですって、［sample1.txt］が消えてしまったので心配しているのですか。いえいえ［sample1.txt］はちゃんと存在しています。

画面右上のファイル名の前の番号を見てください。［2］という数字がありますね。これは、「2つめのファイルを表示しているよ」という意味のなのです。ですから、1つめのファイル［sample1.txt］はどこかにあるはずなのです。ここで **^KW**（**[f･2]**）を打ってください。画面に［sample1.txt］が現れたでしょう。もう一度 **^KW**を打つと、画面に［sample2.txt］が戻ってきます。お分かりでしょうか、**^KW** は、扱っているテキストを交換するコマンドなのです。

では3つ以上のファイルを同時に処理しているときはどうすればよいでしょうか。その時のために **^KT**（**[f･3]**）を説明しておきます。

```
┌【編集テキスト】─────────────┐
│ 1  sample1.txt              xxxxxx │
│ 2  sample2.txt              xxxxxx │
└──────────────────────┘
```

このようにファイルの名前の一覧表を表示しますから、カーソルで選んでリターンキーを押してください。カーソルを動かすのが面倒な方はファイル名の前に書いてある数字のキーを打ってもかまいません。ファイル名の後ろの数字は、そのファイルのバイト数です。

[ESC]O は、現在処理しているファイルの処理はそのままで、新しいファイルをオープンする、と憶えておいてください。

VZエディタは標準設定で10個までのファイルを同時に処理することができます。**[ESC]O**を使ってマルチ・ファイル・エディタの威力を楽しんでください。

でも、2つのテキストを同時に見たいときもありますね。それには、**^KS**（**[f･4]**）を押します。画面が上下2つに分かれましたね。カーソルは上のテキストのところにいます。ここで **^KW**（**[f･2]**）を押すとカーソルは下のテキストに移ります。両方のテキストを見ながら、好きな方を編集できるのです。

もう一度 **^KS** を押してください。今度は左右に分かれましたね。ここでも、**^KW** を使って好きな方のテキストにカーソルを移すことができます。

さらに ^KS を押すと、再び一番最初の画面に戻ります。
このように、2つのテキストに対して、

「裏表」→「上下」→「左右」→「裏表」

と画面の分割モードを変えることができるのです。

今までの画面分割は、画面の真中で2つに分かれていました。でも仕事によっては、どちらかの画面を大きく使いたいときもありますね。もう一度 ^KS を押して画面を上下に分割してください。そこで ^KF ([SHIFT]+[f･4]:) を押します。画面の下に

《カーソルキーで位置を移動して[CR]》

という指示が出ます。^X ([↓]) を押してみてください。真中の境界線が下に移動するでしょう。好きなところでリターンキーを押して、境界線を確定してください。

画面を上下や左右に分割しているときは、^KF で境界線の位置を自由に動かすことができます。一方のテキストを広く見たいときには便利ですね。

ウィンドウ切替え ・・・・・・・・ ^KW, [f･2]
編集テキスト切替え ・・・・・・ ^KT , [f･3]
分割モード切替え ・・・・・・・・ ^KS , [f･4]
分割位置の変更 ・・・・・・・・・・ ^KF , [SHIFT]+[f･4]

だいぶ長くなりましたので、ここら辺で一休みしましょう。[ESC]Qを押してください。

《修正テキストを出力しますか? (Y/N)》 に対しては [N]
《エディタを終了しますか? (Y/N)》 に対しても [Y]

これはもう憶えましたね。

二つのファイルの比較

さて、今度は、次のように入力してください。

```
A>vz sample1 sample2↵
```

2つのファイルを一度に読みこみましたね。このように、ファイル名を指定するとき、複数のファイル名を一度に指定できます。各々のファイル名は、スペースで区切ってください。また、拡張子 .txt を省略しています。拡張子を省略した場合は、次にあげる拡張子のつくファイルを順番に探していきます。

.doc .txt .bat .def .h .c .inc .asm

さて、今は［SAMPLE2.TXT］が画面に出ていますね。^B でブロックモードにして、好みの部分をブロック指定し、^KK で記憶してください。そうしたら、^KW で［SAMPLE1.TXT］に移ります。カーソルを好きなところに置いて ^J を押してください。［SAMPLE2.TXT］のブロックの部分が［SAMPLE1.TXT］に移りましたね。

このように「部分の切り貼り」は、別のテキストの間でも可能なのです。この機能を利用すれば、まるでノリとハサミを使うように自由に文書の切り貼りができます。この機能を「カット・アンド・ペースト」と呼びます。この機能のキーを忘れた場合は、ヘルプ・メニューの『B カット&ペースト』を見てください。

複数のファイルを利用する際の2番目の武器は ^QW と ^QX です。これは「スムース・スクロール」でしたね。でも2つのファイルを画面の左右分割で見ているときには、おもしろい機能を持っています。2つのファイルが同期して動くのです。「同期」という意味は、右のファイルが1行動くときに左のファイルも1行動くということです。この機能を利用すると、2つのファイルの比較を目で見ながら行うことができます。

2つのテキストを比較して、違いを探したいのなら、もっと便利な機能があります。^KX（[SHIFT]+[f･3]）です。この機能は、2つのファイルの相違点を自動的に発見するものです。まず2つのファイルで ^QR を使ってカーソルをファイルの先頭に持ってきます。その後で ^KX を打つと、カーソル位置から2つのファイルを比較し、相違点を発見するとカーソルをその位置に持ってきて、

《相違があります》

という黄色のメッセージを画面の一番下に出します。

ファイルメニューの説明

さてそれでは、[f･1]で表示する『ファイルメニュー』の説明の続きにもどります。

■C ファイルのクローズ ([ESC]C)

現在編集中のテキスト(カーソルが出ているテキスト)の編集を終ります。編集中のテキストに少しでも新しい書き込みや修正があった場合には、

《"A:¥SAMPLE2.TXT"を出力しますか?(Y/N)》

と聞いてきますから、ディスクにしまいたいときは[Y]、消えてしまってもよいときは[N]を打ってください。

このコマンドは、同時に処理をしている別のテキストには影響がありません。複数の文書を処理していて、一つだけ編集を終了したいときに便利です。なお、修正を行っているテキストの場合は、書き込みをせずに終了します。

最後のテキストを[ESC]Cでクローズすると、どうなるでしょう。あれっ、『入力ファイル』というウィンドウが出てきました。[ESC]Cで最後のテキストをクローズすると、起動したときの状態に戻るのです。ここでまたファイル名を書けば、編集を続けることができます。DOSに戻る時は、[ESC]を押してください。

■S ファイルのセーブ ([ESC]S)

エディタを使ってプログラムを書いているときなどに停電があったら大変ですね。コンピュータは即死で、せっかく書いた内容も消えてしまいます。そんなことのないように、30分置きぐらいでディスクにしまっておくと良いでしょう。それには[ESC]Sを押してください。

```
【出力ファイル】
A:¥SAMPLE1.TXT
```

のような、青い箱が出て、ドライブ名、ディレクトリ名がきちんと付いたファイル名を表示します。これで良いならリターンキーを押してください。現在編集しているファイルをディスクに書き込んでまた編集モードに戻ります。

このコマンドにはもう一つ便利な使い方ができるのです。テキストの一部をブロックに指定して、それを新しいファイルとしてディスクにしまう機能です。では、テキストの英文の部分を新しいファイルにしてみましょう。1番の英文の最初の文字［1］にカーソルを置いて ^B を打ちます。それから13番の英文のところにカーソルを持っていってください。英文の部分が青くなりましたね。そこで［ESC］Sを打ちます。ファイル名は［sample3. txt］とでもやっておきましょう。リターンキーを打つと、ブロックの範囲を新しいファイル［sample3. txt］にセーブします。

■ N 新規ファイルオープン （[ESC]N）

新しいファイルをオープンします。［ESC］Oでも代用できますので、ここでは省略します。

■ R Read onlyオープン （[ESC]R）

VZエディタは複数の文書を一度に取り扱うことができますから、何かの文書を編集しているときに、他の文書を見ながら作業を行うことができます。でも、その際に参考にしている文書を訂正してしまっては大変です。このような場合には、文書を「read only」モードでオープンしておきます。このモードでオープンしたファイルは、ディスクへの書き込みはできませんので、ファイルの内容が変わってしまう心配がありません。

なお、^KR を打つことによって、この「read only」モードを設定したり解除したりすることができますから、これも憶えておいてください。「read only」モードの場合は、画面の一番上のファイル名の前に［R］という文字がついており、反転部分が青くなっていますのですぐ分かります。

■ L クローズ・オープン （[ESC]L）

現在編集しているファイルをクローズして、新しいファイルをオープンします。［ESC］Cと［ESC］Oを続けて実行するのと同じです。

■ A 既存ファイルへ追加 （[ESC]A）

現在編集中の文書を何かのファイルの後にくっつけたいときがあるでしょう。このようなときに使います。例えば、［bunsho. txt］というファイルが存在しているとします。現在編集中の文書をこの後に追加したいときは、［ESC］Aを打ちます。

```
┌【追加ファイル】──────────────────────┐
│                                        │
└────────────────────────────────────────┘
```

追加先のファイルの名前を聞いてきますから、

bunsho.txt↵

と打ってください。これで文書の追加ができます。パソコン通信のログファイルの整理には欠かせないコマンドですね。

■P ファイル名の変更 ([ESC]P)

現在編集しているファイルを違う名前に変えたいときに使います。[ESC]Sでセーブする際に、ファイル名は変更できますから、あまり使わないでしょう。

■D テキストの二重化 ([ESC]D)

テキストの編集をしている際に、編集前のテキストを参考にしながら作業を行いたい場合があります。[ESC]Dを打つと、瞬時に同じファイルをもう一つオープンし、画面を上下に分割して表示します。新しくオープンしたファイルは「read only」モードになっていますから、トラブルは発生しません。

■U 編集のやり直し ([ESC]U)

現在編集中のテキストについて、編集を始める前の状態（ファイルをオープンしたときの状態）に戻します。

■I カーソル位置への挿入 ([ESC]I)

現在編集中のテキストに、何か別のファイルを読み込みたいときがあります。このような場合に使います。読み込みたい場所にカーソルを動かして、[ESC]Iを打ちます。

```
┌【入力ファイル】<A:¥>──────────────────┐
│                                        │
└────────────────────────────────────────┘
```

というウィンドウが開きますから、読み込みたいファイル名を入力してください。読み込んだファイルの内容は、カーソル位置の後に入ります。

■Q エディタを終了 ([ESC]Q)

これは以前に何回も使いました。複数のファイルを処理している際は、修正または追加を行ったファイルが一つでもあると、

《修正テキストを出力しますか? (Y/N)》

と聞いてきます。[**Y**]と答えると、全ての修正したファイルをセーブし、[**N**]と答えると、何もセーブしません。

■X 全ファイルのクローズ ([ESC]X)

複数のテキストを編集している時に、それぞれのテキストを[**ESC**]**C**でクローズするのと同じ動作をします。

■E DOSコマンド実行 ([ESC]E)

[**ESC**]**E**を押すと、VZエディタから離れること無しにMS－DOSのコマンドを実行できます。画面はVZエディタを起動したときのMS－DOSの画面に戻り、任意のMS－DOSのコマンドを実行することができます。

この際に、これまで説明してきたカーソルの移動や削除などの編集キーを使うことができます。通常のMS－DOSの上では、編集は行えませんので非常に便利です。

また、このモードで実行したコマンドはすべて記憶してあります。ですから、以前に実行したコマンドを呼び出したいときには、^**E** ([↑]) を打ってください。カーソル位置に次々と前に入力したコマンドが現れます。古い方へ進むときには ^**E** ([↑])、新しい方へ進むときには ^**X** ([↓]) を押します。この機能を使うと、一度実行したコマンドや似たようなコマンドを使うときは、これを呼び出して好きなように書き換えて使うことができます。その代わり、MS－DOSのテンプレート機能は使うことができません。

いろいろなモード

［f･1］のメニューの説明がひととおり終わったところで、もう一つのメニューの説明に入ります。実は、VZには、標準で3つのメニューが設定されています。残りのメニューは、［SHIFT］+［f･2］で開きますが、このメニューの説明は、本章では省略いたします。

［SHIFT］+［f･1］を押すと、いろいろなモードのメニューが開きます。カーソルを必要な項目に持っていってリターンキーを押すと、モードが切り替わり、画面が再表示されます。ただし、「画面の横幅」や「ページ長」のように数字で設定する項目だけは、数値入力用のウィンドウがさらに開きます。設定が終ったら、［ESC］を押してウィンドウを閉じてください。項目の前の英字を入力すると、モードが切り替わって、ただちにウィンドウを閉じます。

特によく使う項目について、簡単に説明しましょう。

■C CR の表示
■T TABの表示
■E EOFの表示

青色の矢印で表示している、改行マーク、タブマーク、ファイルの終りのマークなどを表示するかどうかを設定します。

■L 行番号の表示

画面の左端に行番号を表示するかどうかを設定します。

■N 行番号モード

VZエディタでは、通常は、改行マークから改行マークまでを1行と考えています。これを「論理行」と呼びます。しかし、印刷のための編集などで、1行の字詰めを考えるときには、画面表示の行を1行と考える便利です。この場合の「行」を「表示行」と呼びます。スティタス・ラインに表示されている行番号は、通常「論理行」で数えた行番号ですが、このメニューで「表示行」で数えた行番号に変更できます。こうすると、カーソルのジャンプが多少遅くなります。

■A オートインデント

[ON] にしておくと、編集モードでリターンキーを押したとき、カーソルが一つ上の文の先頭と同じ位置に移動します。同じ字下げの文を続けて書くときに便利です。

■I 検索時の英大/小文字

英字の単語などを検索する際に、大文字と小文字の違いを無視するかどうかを設定します。

■H ページング単位

^R や ^C のスクロールの際に、前画面をスクロールするか、画面の半分だけスクロールするかを設定します。

■B .BAK ファイルの作成

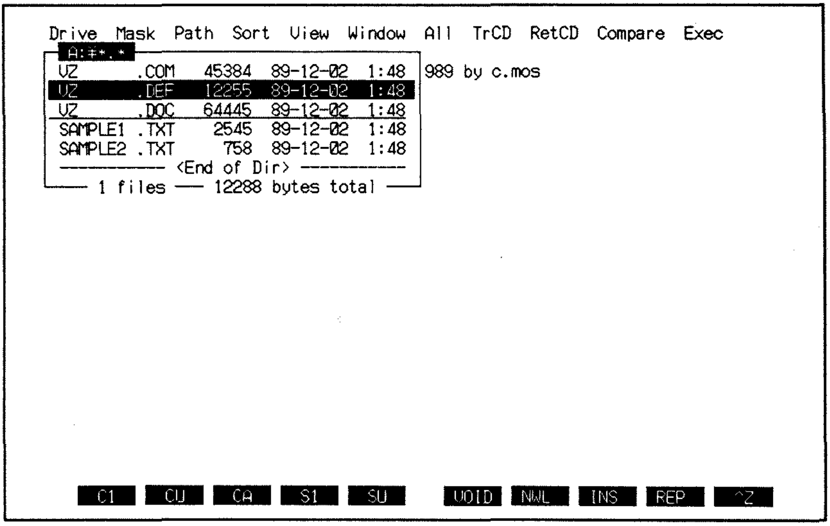
VZエディタは、編集したファイルをディスクにしまう際に、編集前のファイルに [BAK] という拡張子を付けてとっておきます。これをバックアップ・ファイルといいます。この機能を必要としない場合に [OFF] にします。通常は [ON] にしておいた方が安全です。

■ R 画面の横幅(2-250)

画面の表示幅を設定します。表示幅が80を越えた場合、カーソルを画面の左端より左に動かすと、画面が左右に自動的にスクロールします。

ファイラーについて

ファイル名を入力するウインドウで、ただリターンキーを打つと画面ががらりと変わります。これをファイラーモードといいます。ファイラーにはいろいろ便利な機能が有るのですが、ここでは、ごく基本的な機能だけを説明します。詳しく知りたい方は第5章を参照してください。



^X と ^E ([↓]と[↑]) でカーソルを移動してリターンキーを押すとファイルを読み込むことは、前に説明しましたね。実は、複数のファイルを一度に読み込むこともできます。[**SPACE**]を押してみてください。カーソル位置のファイル名が反転しました。反転したファイルは選択されていることを表します。もう一度[**SPACE**]を押すと、もとに戻ります。このようにして、複数のファイルを選択し、リターンキーを押すと、選択されたファイルを一度に読み込みます。

表示するディレクトリを変更したいときには、ディレクトリのところへカーソルを持っていってリターンキーを押してください。そのディレクトリを表示します。このよう

にして、他のディレクトリのファイルを読み込むこともできます。

それではドライブ名を変えたいときはどうすれば良いでしょうか。

上の図で、①のところに書いてあるのは、ファイラーで使えるコマンドの一覧です。このなかに Drive というのが有りますね。これがドライブ名を設定するコマンドです。では［D］を押してみましょう。

```
Drive  Mask  Path  Sort  View  Window  All  TrCD  RetCD  Compare  Exec
       .*
 A:    ------ <End of Dir> ----------
 B:    files ── 1250304 bytes free ──
 C:
 D:
 E:
```

左上にドライブ名を入力する小さなウインドウが開きます。ここでドライブ名を表す英字を打ち込んでも良いですし、カーソルを必要なドライブ名のところへ動かして、リターンキーを押してもよいです。選択したドライブに切り替わります。

ちょっと便利な機能

さてこれで、ひととおりの説明が終わりました。どうもお疲れさまです。ここまで読んでくださったお礼に、知っておくと便利な機能を2、3紹介いたします。

■ 常駐モード

VZエディタを起動する際に、

```
A>vz -z↵
```

のように-zを付けると、VZエディタがメモリ上に常駐します。こうしておくと、[ESC]を打つだけで瞬時にVZエディタが起動します。編集しながらMS－DOSのコマンドもよく使うという方は、是非ともご利用になってみてください。もう一度 A>vz -z と入力すると、メモリを解放します。

VZエディタが常駐している状態では、[ESC] Eのところで説明した機能をすべて使うことができます。このように、VZエディタはMS－DOSのツールとしての機能もいろいろ備えています。この常駐機能については、第7章で説明します。

■ 文字列の記憶

編集モードで ^**L**（[**f･5**]）を押すと、カーソルの位置の単語を記憶します。記憶した文字列は画面左下に黄色で表示しますからすぐ分かります。

実際に使ってみましょう。[SAMPLE1.TXT] の [quick] の [q] の上にカーソルを置いて ^**L** を打ってください。左下に [quick] と表示されましたね。次に好きなところにカーソルを持っていって ^**QI**（[**SHIFT**]+[**f･5**]）を打ってください。[quick] がカーソル位置に現れます。連続して ^**QI** を打つと、どんどん [quick] が現れます。

今度は、^**KA**（[**SHIFT**]+[**f･7**]）と打ってください。

```
┌【複写文字列】──────────────────────┐
│ quick                              │
└────────────────────────────────────┘
```

というウィンドウが開きました。リターンキーを押すと、この [quick] がカーソル位置に現れます。文書やプログラムを書いているときに、何回も出てくる言葉はVZエディタに憶えさせておくと便利です。

また、^**KA** でウィンドウが開いたときに、表示されている文字列を書き変えることもできます。ですから長い言葉を記憶させることもできるのです。もう一度 ^**KA** を打って [quick] を [quick and clever] に直してみましょう。直し終ったらリターンキーを押します。長い文字列でも編集画面に持ってくることができましたね。

ではもう一度 ^**KA** を打ってください。『複写文字列』のウィンドウが開いたら ^**E**（[**↑**]）を打ってください。いろいろな文字列が出てきましたね。実は、検索や置換の際に打ち込んだ文字列も同じところに憶えているのです（そのほかに、あらかじめいくつかの文字列が入っています）。従って、^**L** でカーソル位置の文字列を記憶して、^**C**, ^**R** で検索することもできます。

■ キーボード・マクロ

一連の同じ操作を何回もやらなければならないときがよくあります。このような場合、その一連の操作を一つのキーに登録できれば非常に便利ですね。いわば即席のファンクション・キーです。

仮に文書を書いているとしましょう。出来上がった後で小見出しを付ける際などで、

[↵] ★★ [↵] [↵]

という操作を何回もやることになります。これをなにか空いているキーに登録してみましょう。VZエディタで空いているキーは ^¥ や ^O です。

まず、^_（キーボードの右下にあるキーです）を打ちます。画面左下に黄色い文字で [m] を表示していますね。これは、マクロの登録中という印です。次に、

[↵] ★★ [↵] [↵]

と順に打っていきます。画面にキー操作の結果が出てきますが気にしないでください。最後に ^¥ を打ってください。画面左下に [OK!] と表示されました。これで登録終了です。

以後、^¥ を打つだけで

↓
★★↓
↓

を任意の場所に書き込むことができます。なお、この登録を消すには、^_ を打ってから ^¥ を打ってください。これで、^¥ を打ってもなにもしなくなります。途中で登録を中断する時は、もう一度 ^_ を打ちます。

この機能をキーボードマクロと呼びます。詳しくは第11章をご覧ください。

第2章 VZの起動

2.1 コマンドラインの書式

2.2 環境変数

2.3 編集ファイル名の省略

2.4 動作環境の設定

2.1 コマンドラインの書式

VZを起動する際は、コマンドラインでいくつかのパラメータを指定することができます。

[例]

(1)　A>vz↵

(2)　A>vz vz.def vze.def↵

(3)　A>vz -an6↵

(4)　A>vz /mi↵

(5)　A>vz +game↵

(1)のように何も指定しない場合は、

```
┌【入力ファイル】<A:¥> ──────────┐
│                                │
└────────────────────────────────┘
```

というウィンドウが開きますから、ここで編集ファイル名を入力します。(2)のように直接ファイル名を指定することもできます。複数ファイルを指定する場合は、スペースで区切ります。

(3)はオプションを指定する場合です。オプションは、このように「-」に続いて指定します。オプションは、必ずファイル名の前に指定しなくてはなりません。
この例の「-**an**6」は、文字色を黄色（J-3100版ではボールド）にするオプションです。

(4)は「/」に続いてDEFファイル名を指定しています。VZは、起動する際にコマンドキー割り当てやオプション、マクロ等、変更できる内容をまとめたファイルを必ず読み込みます。これは拡張子「.def」のファイルで、これらをDEFファイルと呼びます。標準では「vz.def」を読み込みますが、このように他のファイルを指定することで、それぞれ違った操作環境で起動することができます。(5)もDEFファイルの指定ですが、「+」の場合は、そのファイルに追加して読み込むことになります。マクロを記述したDEFファイルを組み込む際に使用します。詳しくは『第11章 カスタマイズ』をご覧ください。

コマンドラインパラメータの書式をまとめると、次のようになります。

[書式]

A>vz {-<ｵﾌﾟｼｮﾝ>} [/<DEFﾌｧｲﾙ>]{+<DEFﾌｧｲﾙ>} [<編集ファイル指定>]

編集ファイル指定

編集ファイル指定では、複数のファイル名を直接指定する他にも、いくつかの指定の方法があります。
[例]

(1) vz↵ vz.doc -100↵
(2) b:↵ *.txt↵ c:¥doc↵
(3) @↵ @@↵

(1)は通常の直接ファイル名指定です。ディレクトリや拡張子の指定は省略することができますが、これについてはこの章の後半で説明します。「-100」のように、オープン後、カーソルを指定行番号に移動することもできます。
(2)のように、特定のファイル名を指定せず、ドライブ名やディレクトリ名、ファイル名にワイルドカードを使用した場合、または↵のみ入力すると、指定したディレクトリで該当するファイルの一覧を表示します。これをファイラーと呼びます。ファイラーの詳しい操作方法は『第5章』で説明します。とりあえず次のキーで編集したいファイルを選択して下さい。[SPACE]キーで複数のファイルを選択することもできます。

[↑][↓]	:カーソルの移動
[↵]	:編集
[SPACE]	:選択/非選択
[ESC]	:前のモードに戻る

(3)のように「@」を指定した場合は、編集するファイルの情報を含んだファイル(ピックファイルと呼びます)を参照し、その内容に従ってファイルをオープンします。これは、『3.3 プロファイル機能』で詳しく説明します。

編集ファイル指定時の書式をまとめると、次のようになります。

[書式1] 直接ファイル指定
{<ﾌｧｲﾙ名> [-<行番号>]}
[書式2] ファイラー起動
<ﾊﾟｽ名>|<ﾌｧｲﾙﾏｽｸ>|↵のみ
[書式3] ピックファイル指定
@[<ﾌｧｲﾙ名>]

DEFファイル指定

起動時に読み込むDEFファイル名を「/」の後に指定します。標準で読み込むDEFファイルは、

「起動するVZのコマンド名.DEF」

です。例えばJ-3100版の「vzj31.com」の場合は、「vzj31.def」です。また、別のDEFファイルになっているマクロを組み込みたい場合は、「+」の後にマクロのDEFファイル名を指定します。

※ MS-DOS 2.11の場合は「vz.def」に固定です。

DEFファイルを読み込むディレクトリは、MS-DOSのバージョンによって次のように異なります。

MS-DOS2.11	:カレント・ディレクトリ
MS-DOS3.1以降	:VZの起動パス

これ以外のディレクトリから指定DEFファイルを読み込むこともできます。これについては、『2.2 環境変数』で説明しています。

DEFファイルの詳しい内容については、『第11章 カスタマイズ』で説明しますが、まず一度ファイルをご覧になってみてください。

オプション指定

コマンドラインオプションは、「-」に続いて指定します。複数のオプションを指定する場合は、スペースで区切って下さい。書式は次のとおりです。

[書式]

-<ｵﾌﾟｼｮﾝ名>[+|-|n]

<ｵﾌﾟｼｮﾝ名>	:英字1字または2字（大文字、小文字は無視）
+ または無指定	:スイッチON（値=1）
-	:スイッチOFF（値=0）
n	:値=n （10進数）

[例]

```
A>vz -du -dc- -an6 vz.def↵
```

VZは、たいへん多くのオプションを備えています。これらはDEFファイルの終わり近くの「* 0 オプション」以降に、記述してあります。

```
* 0 オプション

Bt64                    ;テキストバッファのサイズ (16..64KB)
Bq0                     ;メモリ上のテンポラリバッファのサイズ (KB)
...
```

常時変更して使用する場合は、DEFファイルのこの部分を直接書き変えてください。オプションの機能について簡単なコメントもありますが、詳しい機能については『第11章カスタマイズ』を参照してください。

コマンドラインで指定した場合は、DEFファイルの設定に優先しますので、一時的に変更して状態をみたい場合にお使いください。

編集時にはオプションの幾つかは、[SHIFT]+[f･1]のメニューで変更できます。また、[ESC]+[@]と打つと、

【マクロ文】

というウィンドウが開きます。ここで

an6↵

などと入力することで、動作中にオプションを変更することもできます。この場合は、先頭に「-」を付けてはいけません。入力後、画面最下行に

《Result: 7 7 $7 ...》

のようにオプションの値を表示します。(ゴミがいっしょに表示されますが、気にしないでください。) オプション名のみを入力すると、変更せずに値を表示できます。

2.2 環境変数

MS-DOSには、「環境領域」と呼ばれる情報格納エリアがあります。まだ御存知ない方は、

A>set↵

を実行してみてください。

```
PATH=
COMSPEC=¥COMMAND.COM
```

などと表示されると思います。ここで「PATH」や「COMSPEC」を「環境変数」と言います。環境変数は、主にユーザーのドライブ使用状況、ハードディスク等のディレクトリ構成によって、MS-DOSが参照する特定のファイルを捜し出すために、該当するパス名等を設定します。例えば、「PATH」はコマンド、バッチファイルの格納場所を、「COMSPEC」は「command.com」の場所を指定します。

環境変数の設定は、

A>set comspec=c:¥command.com

などのように、setコマンドで行ないます。「=」の前後にスペースを入れてはなりません。

★注意★

VZから[ESC]Eでコマンドラインモードへ移り、そこでsetコマンドを実行しても、環境変数は変更されません。(常駐時は可能です)

VZは次の4つの環境変数を参照します。

VZDEF	:DEFファイルのディレクトリ・ファイル名
VZPATH	:編集ファイルのディレクトリ
VZBAK	:バックアップディレクトリ (Ver1.5で追加)
TMP	:テンポラリファイル

★注意★

Ver1.5では、VZのコマンド名に関わらず上記の環境変数を参照します。

■ VZDEF

DEFファイルのディレクトリ、および起動時に読み込むデフォルトのDEFファイル名を指定します。DEFファイル名の拡張子「.def」は省略できます。また、ディレクトリのみ指定する場合は、ディレクトリ名の最後に「¥」を付けてください。

[例]

```
A>set vzdef=mi.def
A>set vzdef=c:¥def¥mi
A>set vzdef=c:¥def¥
```

■ VZPATH

編集ファイル指定でディレクトリを省略した場合に、ファイルを捜すディレクトリをスペースで区切って指定します。ディレクトリの個数に制限はありませんが、あまり多く指定すると、誤ってファイル名を入力した際に検索時間がかかります。

ディレクトリ名の最後に「¥」を付ける必要はありません。

[例]

```
A>set vzpath=c:¥h c:¥inc c:¥vz¥def
```

ファイル名の省略指定に関しては、次節で詳しく説明します。

※ 環境変数「VZPATH」は、起動時ではなく、ファイルオープン時に参照します。

■ VZBAK

これはVer1.5で追加された環境変数です。従来は、バックアップモードでファイルをセーブする際に、古いファイルを「.bak」にリネームしてから、編集ファイルをセーブしました。この方法だと、ファイルの並び順序が乱れますし、拡張子のみ異なるファイルだと、重複してしまいます。そこで、Ver1.5から、「バックアップディレクトリ」を取り入れました。これは、環境変数「VZBAK」に、あらかじめバックアップファイルを格納するディレクトリ名を設定しておくことで、セーブ時に、古いファイルをバックアップディレクトリの方へ移す機能です。

例えば、次のように設定します。

```
A>set vzbak=¥trash
```

そして、ドライブCのファイルを編集します。

```
A>vz c:¥file.txt
```

[**SHIFT**]+[**f・1**]でバックアップモードがONになっていることを確認し、[**ESC**]**S**でセー

ブしてみて下さい。終了後「dir」を実行すると、ドライブCに「¥trash」というディレクトリが作成され、そこへ古いファイルが移動していることがわかります。

```
c:¥file.txt           : セーブしたファイル
c:¥trash¥file.txt     : 古いファイル
```

バックアップディレクトリへ移動するか、従来の「.bak」にリネームするかは、環境変数「VZBAK」が設定されているかどうかによります。また、バックアップディレクトリは、ドライブごとに作成されます。これは実際にファイルをコピーするのではなく、ディレクトリ上で移動させているためです。従って、バックアップディレクトリ名にドライブ名を加えてはなりません。

※ 環境変数「VZBAK」は、起動時ではなく、ファイルセーブ時に参照します。

■ TMP

大きなファイルを編集する際には、テンポラリファイルを作成する必要があります。このテンポラリファイルを作成するドライブ・ディレクトリを指定するのが環境変数「TMP」です。これを設定しない場合、起動時のカレントディレクトリへ作成します。

RAMディスクやハードディスク等の高速で大容量のドライブをお使いの場合は、そのドライブを割り当ててください。

[例]

```
A>set tmp=c:¥tmp
```

テンポラリファイルを作成するドライブには、十分なフリーエリアが必要です。最低でも、編集するファイルサイズからメモリ上に読み込むサイズ（オプション ***Bt*** で指定）を引いたサイズを、同時に読み込む数だけ合計したエリアが必要となります。フロッピーディスクのみで使う場合等、テンポラリファイルを作成するエリアがない場合には、「オプション ***Bq***」を指定することで、メモリ上にテンポラリファイルを作成することができます。　　【 ***Bq*** 0 :メモリ上のテンポラリバッファのサイズ（KB）】

[例]

```
A>vz -bq256      (256KBまでメモリ上にテンポラリファイルを作成)
```

いずれにしろ、テンポラリファイル領域が十分でない場合は、編集ファイルを読み込んだ後、まず **^QC** でテキストを最後まで読み込み、再び **^QR** で先頭に戻って、

《テンポラリファイルエラー》

が表示されないことを確認してから、編集を開始してください。また、セーブする際は

バックアップモードはOFFにしてください。
Ver1.5では、大きなファイルを眺めるだけで編集しない場合、テンポラリファイルを作成しない「Viewモード」を追加しました。このモードは、^QVで変更できます。これについては、『3.1 ファイル入出力』で触れます。

※ 環境変数「TMP」は、起動時のみ参照します。

2.3 編集ファイル名の省略

編集ファイル名を指定する際には、ディレクトリ名、拡張子は省略できます。この機能はVer1.5で大幅に拡張されました。

■ ディレクトリの省略

ディレクトリを省略した場合は、まずカレントディレクトリで一致するファイルを捜し、存在しない場合は環境変数「VZPATH」で指定したディレクトリを順番に捜していきます。Ver1.5では、拡張子を省略した場合も、ディレクトリのサーチを行ないます。

■ 拡張子の省略

拡張子を省略した場合は、次の順序でファイル名が一致するファイルを捜します。

1.拡張子がないファイル
2.デフォルト拡張子をもつファイル
3.デフォルトバイナリ拡張子でないファイル

「デフォルト拡張子」は、DEFファイルの「* E その他 3」に指定した拡張子です。

```
3       .doc .txt .bat .def .H .C .inc .asm
```

この中の拡張子をもつファイルが複数ある場合は、前に指定した拡張子のファイルを優先します。

「デフォルトバイナリ拡張子」は、Ver1.5で追加された項目で、DEFファイルの「* E その他 9」で指定します。

```
9       .com .exe .obj .lib .arc .lzh
```

デフォルト拡張子のファイルが見つからない場合で、これらの拡張子でないファイルが存在する場合は、それらのファイルのうち、最後に見つかったものをオープンします。

例えば、次のファイルがディレクトリに存在するとします。

```
vz.com vz.def vz.doc pds.man pds.ish pds.com
```

拡張子を省略して指定した場合、

vz → vz.doc
pds → pds.ish

を、それぞれオープンすることになります。
※「* E その他 9」を削除するか、行頭に「;」を挿入してコメントにすると、デフォルトの拡張子のみ捜します。

2.4 動作環境の設定

ここでは、動作環境に合わせたオプション、環境変数の設定例をいくつか示します。

■ フロッピーディスクのみの場合

フロッピーディスクのみの場合は、コマンドの読み込み等のディスクアクセスに時間がかかりますので、まずVZを常駐されることをお勧めします。「常駐モード」に関しては『7.1 常駐モード』で詳しく説明しますが、とりあえず、

```
A>vz -z
```

でメモリに常駐します。いったん常駐すれば、もうVZの起動ディスクは必要ありません。編集を始める時は、[**ESC**]を押してください。

さらに、大きなファイルを編集する際は、メモリ上にテンポラリファイルの領域を確保するために、

【 ***Bq*** 256 :メモリ上のテンポラリバッファのサイズ (KB) 】

としてください。この場合は256KBまでメモリを使用します。

■ RAMディスクがある場合

RAMディスクがある場合は、以下のファイルをRAMディスクに転送してください。

```
vz.com   vz.def   command.com
```

そして、次のように環境変数を設定します。(RAMディスクがドライブCの場合)

```
A>set comspec=c:¥command.com
A>set tmp=c:¥
```

これらの処理を次のように起動ドライブの「autoexec.bat」に記述しておくと、起動時に自動的に実行できます。

```
set %RD%=c:
if exist %RD%¥command.com goto skip
copy command.com %RD%
copy vz.* %RD%
:skip
```

```
set comspec=%RD%¥command.com
set tmp=%RD%¥
```

ここではRAMディスクのドライブ名を環境変数「RD」に設定し、RAMディスクのドライブが変わった場合には、1行目を変更するだけでよくなっています。また、RAMディスクが不揮発性のもので、リセットしても内容が消えない場合は、すでにあるものをコピーする必要はありませんので、2行目でRAMディスクにファイルがあるかどうかチェックしています。

■ ハードディスクがある場合

ハードディスクがある場合もRAMディスクの場合と同様ですが、RAMディスクの方がはるかに高速ですから、容量が十分ある場合には、テンポラリファイルはRAMディスクに作成しましょう。

また、ハードディスクで作業ディレクトリがたくさんある場合は、環境変数「VZPATH」に作業ディレクトリを設定してください。そうすると、ディレクトリ指定を省略してファイルをオープンできます。

```
A>set vzpath=c:¥vz c:¥doc c:¥tc c:¥tc¥h
```

■ 裏グラフィックVRAM

PC-9801のノーマルモードでは、なるべく裏グラフィックVRAMを使用してください。メインメモリの消費が64KB少なくてすみます。これは「オプション***GV***」で指定します。

【***GV***1　:裏VRAM使用モード 】

裏グラフィックVRAMを使用するアプリケーションを動かす場合は、試しに「***GV***3」に設定して、それで支障がないようならその設定でお使いください。

■ EMSを利用できる場合

EMSを利用できる場合は、次のオプションを付けて常駐してください。

```
A>vz -sw3 -z
```

常駐サイズが2.5KB弱で済みます。また、VZ以外にもEMSを利用するアプリケーションを使う場合には、VZが使用するEMSのページ数を設定します。

```
A>vz -sw3 -em8 -z
```

最低でも8ページは確保してください。それ未満の場合は、「-***sw***3」は指定しないでください。

■ 日本語FEPの制御

VZは、標準設定のままでもVJEβ、松茸、ATOK6/7を自動判別して、ON/OFFを制御します。ただし、FEPを「I/Oバンク RAM」上で動作させる場合は、自動判別ができない場合がありますので、次のオプションで使用するFEPを設定してください。

【***FT*** :FEPのタイプ (0=自動識別, 1=vjeb, 2=mttk, 3=atok)】

また、編集モードに入った時点で自動的にFEPをONしたい場合は、「オプション***FR***」を設定します。

【***FR*** :日本語入力モードで起動 】

この時、引数としてFEPの初期モードを指定できます。

[例] ATOKの場合

FR1 : ローマ字漢字
FR2 : カナ漢字
FR3 : 半角英数カナ

★注意★

ATOK7をEMS上で使用している場合は、必ず「7to6.com」を常駐してください。

■ アプリケーションからVZを呼び出す場合

dBXL等のアプリケーションからVZを呼び出す場合で、動作メモリが限定される時は、次のようにオプションを設定してワークエリアを減らしてください。

【***Bt***16 :テキストバッファのサイズ 】
【***Bq***0 :メモリ上のテンポラリバッファのサイズ 】
【***Bf***0 :DOSコマンド用メモリの最小値 】
【***Bl***256 :行バッファサイズ 】
【***Bm***256 :マクロバッファのフリーエリア 】
【***TC***2 :オープン可能なテキスト数 】
【***FW***128 :ディレクトリバッファの数 】

また、裏GVRAMを使用できない場合は、次の指定も必要です。

【***Bo***0 :コンソール出力取込みバッファサイズ 】

添付の「vzs.com」を使用すると、ファイラー等の環境機能は使えなくなりますが、より少ないメモリで動作します。

第3章　ファイルの操作

3.1 ファイル入出力

3.2 マルチファイル

3.3 プロファイル機能

3.1 ファイル入出力

この章では、編集ファイルのオープン、クローズ、VZの終了方法、編集ファイルの切り替え、ウィンドウ操作など、ファイル関連のコマンドについて、一通りの説明を行なっていきます。まず、ファイル入出力操作のコマンドから初めましょう。

[f•1]を押すと、次のメニューを表示します。

```
【ファイルメニュー】
O ファイルのオープン
C ファイルのクローズ
S ファイルのセーブ
N 新規ファイルオープン
R Read onlyオープン
L クローズ•オープン
A 既存ファイルへ追加
P ファイル名の変更
D テキストの二重化
U 編集のやり直し
I カーソル位置への挿入
X 全ファイルのクローズ
E DOSコマンド実行
Q エディタを終了
```

これらのコマンドは、メニューの先頭の英字キーを使って、

[ESC][英字]または[GRPH]+[英字]（98版以外では[ALT]+[英字]）

でダイレクトに実行することもできます。では、それぞれのコマンドについて説明していきます。

★注意★

98版で[GRPH]+[英字]を使う場合は、あらかじめ付属の「ezkey.com」を常駐しておく必要があります。

[**ESC**]**O**（Open）　　　:ファイルのオープン

編集テキストをオープンします。コマンドを実行すると、

```
┌【入力ファイル】<A:¥> ──────────────┐
│                                    │
└────────────────────────────────────┘
```

というウィンドウが開きますので、前章で説明したように編集ファイルを指定してください。指定したファイルが見つからない場合には、

《・・・・は見つかりません。新しいファイルですか？（Y/N)》

と聞いてきますので、

[**Y**]　　　　　　:新規ファイルとしてオープン
[**N**],[**ESC**]　　:再び入力ファイルを指定

で答えます。ただし、「オプション***Sn***」を「-」に指定している場合には、この問い合せは無く、ただちに新規ファイルとしてオープンします。

【 ***Sn+*** :新規ファイルの確認 】

指定したファイル名が、すでにオープンされているテキストのいずれかと一致する場合には、

《・・・・はオープンされています》

と表示し、そのテキストが編集テキストになります。

●●　同時にオープン可能なファイル数　●●

同時にオープン可能なファイルは、標準設定で最大10個です。
これは「オプション***TC***」で設定できます。(100個以上も可能です)

【 ***TC***10 :オープン可能なテキスト数 】

これ以上のファイルをオープンしようとすると、

《オープンできません》

と表示します。また、読み込んだテキストは、「オプション***Bt***」で指定したテキストバッファのサイズ分だけのメモリを占めますから、メモリ上の制約でオープンできない場合もあります。たくさんのテキストを同時に編集したい場合は、

【 ***Bt***16 :テキストバッファのサイズ (16..64KB) 】

のようにバッファのサイズを小さくしてください。さらに、EMSメモリを使用できる場合は、より多くのテキストをEMSメモリ上に読み込むことができます。

●●　ファイル名の表示　●●

ステータスラインの右端の編集ファイル名では、カレントドライブ・ディレクトリと一致する部分は省略します。これをフルパス名で表示したい場合は、「オプション***Df***」を「+」に設定してください。

【 ***Df***- :フルパス名で表示 】

ただし、表示できる桁数に収らない場合は、適当な位置から表示します。

実行中にカレントドライブ・ディレクトリを変更しても、すでにオープンされているファイルの表示形式は変りません。 ファイル名はすべて大文字で表示しますが、「オプション***Dp***」を「+」に設定すると、小文字表示になります。

【 ***Dp***- :ファイル名の小文字表示】

[ESC] R (Read)　　　　:Read onlyオープン

編集テキストをRead onlyモードでオープンします。
Read onlyモードのテキストは、ステータスラインが水色で表示され、ファイル名の前に「R」を表示します。この状態のテキストは、修正しても終了時にセーブを確認しませんし、セーブすることもできません。Read onlyモードは、^**KR**コマンドで通常モードに戻せます。Read onlyオープンでは、編集中のテキストと同じファイル名を指定した場合も、重複してオープンします。

●●　修正フラグの表示　●●

編集ファイル名の前には、次のような修正フラグを表示します。

無表示　:修正されていない
*　　　:修正された
R　　　:Read only
V　　　:Viewモード

テキストに編集操作を行なっても、その時点では修正フラグは立ちません。カーソルを他の行へ移動したり、行の編集以外のコマンドを実行した時点で、テキストが変更されている場合に修正フラグが立ちます。すなわち、誤って1文字入力しても、即座に[**BS**]で元に戻せば、修正されたことにはなりません。

「V」は、Ver1.5で追加されたViewモードの表示です。Viewモードは、この節の後半で説明します。

[**ESC**] **N** (New)　　　:新規ファイルオープン

新規ファイルをオープンします。コマンドを実行すると、

【新規ファイル】

というウィンドウが開きますので、新しく作成したいファイル名を入力してください。指定したファイルがすでにディスクに存在する場合は、

《····はすでに存在します》

と警告しますが、テキストはそのままオープンします。[**ESC**] **O**コマンドで存在しないファイルを指定しても新規ファイルを作成できますが、その場合はファイルサーチに時間がかかります。

[**ESC**] **C** (Close)　　　:ファイルのクローズ

編集中のテキストをクローズします。テキストが修正されている場合には、

《････を出力しますか？（Y/N)》

と聞いてきますので、

[Y]　　　　：現テキストをセーブした後、クローズ
[N]　　　　：編集内容を放棄してクローズ
[ESC]　　　：コマンドの実行を中止

で答えてください。

最後の編集テキストをクローズすると、前回テキストをオープンした方法により、

・ファイラーから選択した場合 ･･････････ ファイラー
・直接ファイル名を指定した場合 ････････ 【入力ファイル】ウィンドウ

に戻り、続けて次のファイルをオープンすることができます。この機能を使わず、ただちに編集を終了したい場合は、次のオプションを「+」に設定してください。

【 **Sq**⁻　:Close, Quitで即座に終了 】

★注意★
[ESC]Cで編集を終了した場合は、『3.3 プロファイル機能』で述べるプロファイル情報は、セーブしません。

[ESC]L (Load)　　　　:クローズ･オープン

編集中のテキストをクローズし、続いて新たなファイルをオープンします。[ESC]Cコマンドと[ESC]Oコマンドを続けて実行するのに相当しますが、次の点が異なります。

・指定したファイルが見つからない場合には、新規ファイルとしてオープンします。
・指定したファイル名が編集中のテキストと一致する場合でも、そのままオープンします。

またVer1.5から直前にファイラーからオープンした場合には、本コマンドでも[ESC]Cと同様にファイラーに戻ることができます。すでにファイルをオープンしている時に、ファイラーで次々にファイルを見たい場合に便利な機能です。

[ESC]S (Save)　　　　:ファイルのセーブ

編集中のテキストまたはブロックを、指定したパス名でセーブします。ブロックモー

ドかどうかにより処理が違います。

■　非ブロックモード

コマンドを実行すると、

```
┌【出力ファイル】─────────────┐
│A:¥README.DOC                │
└─────────────────────────────┘
```

というように、編集中のテキストのフルパス名を表示しますので、必要ならばパス名を変更してセーブしてください。この後の編集テキストのパス名は、変更したものに変ります。

■　ブロックモード

コマンドを実行すると、

```
┌【出力ファイル】─────────────┐
│                             │
└─────────────────────────────┘
```

というウィンドウが開きますので、セーブするブロックのファイル名を入力してください。指定したファイルがすでにディスクに存在する場合は、

《••••はすでに存在します．出力しますか？（Y/N)》

と聞いてきますので、

[**Y**]	：既存ファイルの上にセーブ
[**N**], [**ESC**]	：コマンドの実行を中止

で答えてください。

●●　バックアップファイル　●●

編集ファイルをセーブする場合は、ディスク上のもとのファイルをバックアップします。これは、『2.2 環境変数』で説明したように、環境変数「VZBAK」が設定されているかどうかにより、

・VZBAKを設定 ……………… バックアップディレクトリへ移動
・VZBAKは設定せず ………… 「.bak」にリネーム

の2通りの方法が選べます。また、バックアップをしない場合は、

[**SHIFT**]+[**f・1**] 『 B .BAK ファイルの作成 』
【 ***Eb+*** :バックアップファイルの作成 】

をOFFまたは「-」にしてください。

[**ESC**] **A** (Append)　　　:既存ファイルへ追加

編集中のテキストまたはブロックを、指定したファイルの後に追加します。コマンドを実行すると、

【追加ファイル】

というウィンドウが開きますので、追加したい既存ファイルのパス名を入力します。指定したファイルが存在しない場合には、

《・・・・は存在しません．新たに作成しますか？ (Y/N)》

と聞いてきますので、

[**Y**]　　　:新たなファイルを作成
[**N**], [**ESC**]　　　: コマンドの実行を中止

で答えてください。なお、Ver1.5では、本コマンドでもファイラーからファイルを選択できます。また、ディレクトリ、拡張子を省略して、追加したいファイルをサーチすることも可能です。

[**ESC**] **Q** (Quit)　　　:エディタを終了

エディタを終了します。編集中のテキストのいずれかに、修正したテキストがある場合には、

《修正テキストを出力しますか？ (Y/N)》

と聞いてきますので、

[Y]	：全ての修正テキストをセーブ
[N]	：編集内容を放棄
[ESC]	：編集を続ける

で答えてください。この後再度、

《エディタを終了しますか？ (Y/N)》

と確認しますので、

[Y]	：エディタを終了
[N], [ESC]	：編集を続ける

で答えます。これがわずらわしい場合は、[ESC] C で触れた「オプション ***Sq***」を「+」に設定してください。

本コマンドでエディタを終了した場合、『3.3 プロファイル機能』で述べるプロファイル情報を、セーブします。

[GRPH]+[X]	:強制終了（マクロ）
[Alt]+[X]	(J-3100, IBM PC)

エディタを強制終了します。編集中のテキストのいずれかに、修正したテキストがある場合には、

《編集テキストを放棄しますか？ (Y/N)》

と聞いてきますので、

[Y]	：編集内容を放棄してエデイタを終了
[N], [ESC]	：キャンセル

で答えます。プロファイル情報はセーブします。

[**ESC**] **P** (Pathname)　　:ファイル名の変更

編集中のテキストのファイル名を変更します。コマンドを実行すると、

```
【ファイル名】
A:¥README.DOC
```

というように、編集中のテキストのフルパス名を表示しますので、ここで変更してください。変更したファイル名がすでにディスクに存在する場合は、

《・・・・はすでに存在します》

と警告しますが、ファイル名はそのまま変更します。

本コマンドを実行すると、テキストに修正フラグが立ちます。

---

^KR [**SHIFT**]+[**f・2**] **R**　:Read onlyモード

編集中のテキストのRead onlyモードを設定／解除します。

---

^QV　　:Viewモード（マクロ）

Ver1.5の機能で、ファイルのオープンモードをViewモードに切り替えます。このコマンドは、編集ファイル名入力、ファイラー等いつでも実行できます。実行すると画面最下行に

《View mode》

または

《Edit mode》

と表示します。

●●　View モード　●●

Viewモードは、ファイルを編集せず、ただ表示するだけのモードです。このモードでオープンしたファイルは、編集操作を受け付けません。編集しようとすると、画面最下行に

《変更できません》

と表示されます。編集できないために、大きなファイルでもテンポラリファイルを作成する必要がなく、その分高速な表示が可能です。

いったん^QVでViewモードに切り替えると、再度^QVで編集モード（Editモード）に戻さない限り、オープンするファイルは全てViewモードでオープンします。

Viewモードでオープンしたファイルは、^KRコマンドで通常モードに戻すことができます。ただし、大きなファイルの場合には、ファイルの先頭部を表示していないと、モードを切り替えることができません。いったん通常モードに変更したファイルは、再びViewモードに戻すことはできません。

リードオンリ属性のファイルは、自動的にViewモードでオープンします。

[ESC] D (Duplicate)　　:テキストの二重化（マクロ）

編集中のテキストを新たにViewモードでオープンし、横分割した別のウィンドウに表示します。編集前のファイルを参照したい場合に便利です。もう一度実行すると、参照ファイルをクローズして、全画面モードに切り替えます。ただしこの区別は、分割画面になっているかどうかで調べていますから、二重化した後全画面モードにした場合は、クローズされません。

★注意★

このマクロはVer1.5で変更されました。お使いのDEFファイルのものと差し替えてお使いください。

[ESC] U (Undo)　　　:編集のやり直し（マクロ）

編集中のテキストを放棄し、同じファイル名で再びファイルをオープンし、ファイルの先頭を表示します。テキストが修正されている場合は、

《編集テキストを放棄しますか？ (Y/N)》

と聞いてきますので、

[Y]　　　　　:編集内容を放棄してコマンド実行
[N],[ESC]　　:キャンセル

で答えてください。

[ESC]I (Insert)　　　:カーソル位置へ挿入（マクロ）

指定したファイルを読み込み、カーソル位置へ挿入します。コマンドを実行すると、

```
┌【入力ファイル】<A:¥>────────────┐
│                                  │
└──────────────────┘
```

というウィンドウが開き、画面最下行に

《カーソル位置へファイルを読み込みます》

と表示します。挿入するファイルを指定してください。指定方法は[ESC]Oと同様です。複数のファイルを指定した場合には、順次挿入していきます。

[ESC]X (eXit) :全ファイルのクローズ（マクロ）

編集中の全ファイルを順次クローズし、修正されている場合は、セーブするかどうか確認します。これは、[ESC]Cコマンドを順次実行するのと同じです。従って、コマンド実行後、[ESC]Cコマンドで最後のテキストをクローズした時と同様に、編集を再開できます。また、[ESC]Cコマンドと同様に、プロファイル情報はセーブしません。

3.2 マルチファイル

VZは、標準設定で最大10個のファイルをオープンすることができます。このうちの1つまたは2つのテキストを画面に表示して編集を行ないます。これらは、次に示す4種類のマルチファイル操作コマンドで管理します。いずれもファンクションキーの左側に割り当てられており、簡単に実行できます。

<ファンクションキー>

	[ﾌｧｲﾙ]	[窓換]	[文換]	[窓割]	[記憶]	[検索]	[置換]	[ｶｯﾄ]	[ｲﾝｻｰﾄ]	[ﾌﾞﾛｯｸ]
ｼﾌﾄ	[ﾓｰﾄﾞ]	[ｻﾌﾞ]	[比較]	[窓動]	[複写]	[ﾏﾙﾁ]	[複写2]	[ｺﾋﾟｰ]	[ﾍﾟｰｽﾄ]	[ﾀｸﾞ]

---

^KW ［f・2］（窓換）　　:ウィンドウ切替え

編集ウィンドウを切り替えます。フルウィンドウで2つ以上のテキストをオープンしている場合は裏テキストと入れ替わり、分割ウィンドウで編集している場合はもう一方のウィンドウへカーソルが移動します。

---

^KT ［f・3］（文換）　　:編集テキスト切替え

オープンしているテキストの中から、編集テキストを選択します。コマンドを実行すると、次のようにオープンしているテキストのメニューが現れ、編集中のテキストの位置にカーソルがきます。

```
【編集テキスト】
1* SAMPLE1.TXT        2521
2  SAMPLE2.TXT         758
```

※　ファイル名の右側の数字は、そのテキストのバイト数です。

通常のメニューと同様にカーソルを移動して[↵]で選択するか、または[1]～[9]キー（10番目以降は[A]～[Z]の英字キー）で、ダイレクトに選択できます。もう一度コマンドキー（[f・3]）を押すと、編集中のテキストの次のテキストに切り替わります。

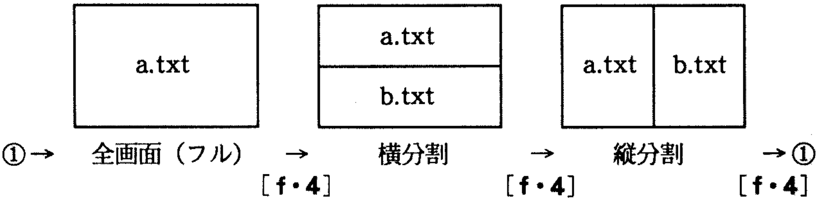
選択したテキストが、裏テキストまたは分割ウィンドウのもう一方のテキストを選択

した場合は、それと入れ替わります。

---

^KS ［f・4］（窓割）　　:分割モード切替え

2つ以上のテキストをオープンしている時、ウィンドウの分割モードを切り替えます。編集テキストは変りません。



---

^KF ［SHIFT］+［f・4］（窓動）:分割位置の変更

分割ウィンドウの分割位置を移動します。コマンドを実行すると、

《カーソルキーで位置を移動して[CR]》

と表示されますので、

・横分割の時　　：カーソル上下移動キー
・縦分割の時　　：カーソル左右移動キー

で分割位置を変更して、[↵]を押してください。

※ ^KL で表示行数を切り替えても、分割位置の割合は保存します。

3.3 プロファイル機能

「プロファイル機能」とは、エディタ終了時に、その時編集していた全テキストのファイル名およびカーソル位置の行番号（プロファイル情報）を、指定ファイルに出力し、起動時にその内容を参照して、ただちに編集を再開できる機能です。プロファイル情報は、「editfile」というファイルに出力され、編集ファイル指定で「@」を入力することで、参照できます。

■ プロファイル情報の書き出し

プロファイル情報は、[ESC]Qでエディタを終了する際にセーブします。ただし、この機能を使用する場合は、あらかじめ「オプション*Sr*」を「+」に指定してください。

【 *Sr*⁻　:editfile にプロファイル情報をセーブ 】

また次のコマンドで、編集時にプロファイル情報をセーブできます。

[ESC]W	:プロファイル情報のセーブ

このコマンドは「オプション*Sr*」の設定にかかわらずプロファイル情報を出力します。

プロファイル情報は、DEFファイルの読み込みパス、およびカレントディレクトリの「editfile」というファイルに出力します。所定のパスに「editfile」が存在しない場合は出力しません。この場合は、あらかじめ次のような方法で、ダミーの「editfile」を作成しておいてください。

[例]

```
A>type nul >editfile
```

また、「editfile」というファイル名を、他のものに変更したい場合は、DEFファイルの

```
* E その他

2       editfile
```

の部分を変更してください。

※ Ver1.5からカレントディレクトリへも出力するようになりました。

■ プロファイル情報の読み込み

プロファイル情報は、編集ファイル指定で「@」を入力することで参照できます。

[例]

```
A>vz @
```

この場合は、DEFファイル読み込みパスの「editfile」を参照します。カレントディレクトリの「editfile」を参照する時は、

```
A>vz @@
```

のように、「@@」と指定してください。(Ver1.5の追加機能です)

プロファイル情報を記録した「editfile」には、

```
A:¥SAMPLE1.TXT -100
A:¥SAMPLE2.TXT -200
```

のように、フルパス名とカーソル位置の行番号が、ファイル数分記録してあります。この型式のファイルを「ピックファイル (Pick file)」と呼びます。この型式のファイルをあらかじめ作成しておき、入力ファイル指定時に「@」に続けて次のようにピックファイル名を指定できます。

@	:DEFパスの「editfile」を参照
@@　@.	:カレントディレクトリの「editfile」を参照
@pickfile	:カレントディレクトリの「pickfile」を参照
@<dir>¥	:<dir>をカレントディレクトリに変更し、そこにある「editfile」を参照
@<dir>¥pickfile	:<dir>をカレントディレクトリに変更し、そこにある「pickfile」を参照

いくつかの作業をサブディレクトリのもとで進めている場合、この機能を利用すると、ただちに該当するディレクトリへ移って作業を再開できます。

※　指定したピックファイルが見つからない場合は、エラーメッセージは出力されず、再度編集ファイルの入力ウィンドウへ戻ります。

第4章　スクリーンエディット

4.1 カーソルの移動
4.2 基本編集コマンド
4.3 ヒストリー機能
4.4 カット＆ペースト
4.5 文字列の操作
4.6 その他のコマンド
4.7 特殊文字の入力

4.1 カーソルの移動

この章では、編集操作に関する全てのコマンドの説明を行ないます。まず初めはカーソル移動に関するコマンドです。これは次のように分類できます。

- ・基本コマンド
- ・ページングモード
- ・スムーススクロール
- ・マークとジャンプ

基本コマンド

カーソル移動の最も基本的なコマンドです。カーソル移動キー、[ROLL UP]、[ROLL DOWN]キーを、[SHIFT]、[CTRL]キーと組み合わせて、キーボードの右側だけで簡単に操作できます。[CTRL]+[英字]、[CTRL]+[Q][英字]キーを利用したものは、^Q[、^Q]を除いて、WordStar系のエディタで共通の操作ですから、覚えておくとたいへん役に立ちます。

コマンドの2行目の(J3100, IBM PC)とあるものは、J-3100版、IBM PC/AX版のキー操作です。

---

^S [←]　^D [→]　^E [↑]　^X [↓]

カーソルを上下左右に1文字（1行）移動します。画面の一番下の行へは移動しません。

---

^A [SHIFT]+[←]　　　:1語←

カーソルを前の単語の先頭へ移動します。

※　単語の区切りの判定方法は後述します。

---

^F [SHIFT]+[→]　　　:1語→

カーソルを次の単語の先頭へ移動します。

---

^QS ［CTRL］+［←］　　　:論理行頭※

カーソルを行の先頭へ移動します。

---

^QD ［CTRL］+［→］　　　:論理行末

カーソルを行の最後へ移動します。

※　改行コードから改行コードまでの1行を「論理行」と呼びます。これに対して画面の1行を「表示行」と呼びます。

---

^Q[　　　　　　　　　　:表示行頭（マクロ）

カーソルを画面の左端へ移動します。

---

^Q]　　　　　　　　　　:表示行末（マクロ）

カーソルを画面の右端へ移動します。

---

^W　［SHIFT］+［ROLL DOWN］　　:ロールアップ
　　［Shift］+［Page Up］　　　(J-3100, IBM PC)

テキストを1行スクロールアップし、カーソルを上の行へ移動します。

---

^Z　［SHIFT］+［ROLL UP］　　　:ロールダウン
　　［Shift］+［Page Down］　　(J-3100, IBM PC)

テキストを1行スクロールダウンし、カーソルを下の行へ移動します。

---

^QE [CTRL]+[↑]　　:画面の上端
[Ctrl]+[Home]　(IBM PC)

カーソルを画面の最上行へ移動します。すでに最上行にある時は、スムーススクロールアップ※を開始します。

---

^QX [CTRL]+[↓]　　:画面の下端
[Ctrl]+[End]　(IBM PC)

カーソルを画面の最下行の1行上に移動します。カーソルが最下行の1行上にある時は、スムーススクロールダウンを開始します。

※　スムーススクロールについては後述します。これらのコマンドによるスムーススクロールの開始は、次のオプションで抑止できます。

【 *Es+*　　:^QE, ^QX でスムーススクロール 】

---

^R [SHIFT]+[UP]　　:ページアップ
[Page Up]　(J-3100, IBM PC)

1ページ前へ戻ります。

---

^C [SHIFT]+[DOWN]　:ページダウン
[Page Up]　(J-3100, IBM PC)

1ページ次へ進みます。

1ページは、次のメニューまたはオプションで全画面または半画面に設定できます。

[SHIFT]+[f･1]　『 H ページング単位 』

【 *Eh-*　　:全／半画面ページング 】

^QR	[CTRL]+[ROLL DOWN]	:ファイルの先頭
	[Ctrl]+[Page Up]	(J-3100, IBM PC)

カーソルをファイルの先頭に移動します。

^QC	[CTRL]+[ROLL UP]	:ファイルの最後
	[Ctrl]+[Page Down]	(J-3100, IBM PC)

カーソルをファイルの最後に移動します。

画面上のカーソル位置の保存

しばらくカーソルを動かしてみるとお気付きなると思いますが、VZでは画面上のカーソル位置を決める際に、カーソルの視点が移動しないようにいくつかの工夫をしています。

■ カラム位置の保存

^E ^X ^W ^Z 等のコマンドで、カーソルが行末以降にくる場合、画面上ではカラム位置を変化させず、行末以降にカーソルを表示します。これは、上下スクロールしても、カーソルが左右に振れて見ずらくならないためです。行末以降にカーソルが表示されても、内部的にはカーソルは行末に存在していますので、エディットコマンド等を実行すると、カーソルは行末にあるものとして処理されます。この機能を禁止したい場合は、次のオプションを「+」に設定してください。

【*En*-　:カーソルを行末以降に表示させない 】

■ 行位置の保存

カーソルを画面の中央にもってきて、^R ^W でテキストの先頭に移動してみてください。先頭行を表示すると、それ以降はカーソルが上に移動してしまいます。これはでは、カーソルを画面の中央に位置させて、^R ^C でテキストを眺めたい場合には、見づらくなります。このため、

(A) 通常はカーソル表示位置の変化を伴わないカーソル移動コマンド
(ロールアップ／ダウン、ジャンプ系コマンド)

(B) その他のカーソル移動を伴うコマンド

という2つのグループにカーソル移動コマンドを分け、(B)グループのコマンド実行時にカーソルが上から何行目にあるかを保存し、(A)グループのコマンド実行時にそれを参照する、という方法を取り入れました。この機能はオプションで禁止することはできません。

ページングモード

通常、^R ^C のページングは画面単位で動作しますが、VZではこれを3つのモードに使いわけることができます。

^@ [SHIFT]+[INS]　　:ページングモード

^R ^C コマンドのページングモードを切り替えます。現在のページングモードは、画面最上部のステータスラインの左端に表示します。

P（表示ページングモード）
　全画面または半画面単位で上下に移動します。

C（論理ページングモード）
　タイトル行をサーチします。(「C」は「Comment」の頭文字です)

S（文字列検索モード）
　^QF, ^L コマンドで設定された文字列を検索します。

各モードは、次の様に切り替わります。
- 検索文字列が設定されていない場合
  P→C→P ...
- 検索文字列が設定されている場合
  P→C→S→P ...

　タイトル行は、行頭のスペース・タブを除いた最初の文字が「* / ; ^L」で、直前が空行（改行のみ）の行です。この行を、サブルーチンの始まり、章の始まりなどに挿入し

ておくことで、ページング機能をより使いやすくできます。
[例]

```
↓
/* Main Routine */↓                ☜ タイトル行
main()↓
...

↓
^L 第1章 VZエディタを...  ☜ タイトル行
...


↓
* M マクロ↓                        ☜ タイトル行
↓
2 [F01] "【ファイルメニュー】"↓
...
```

『4.5文字列の操作』で述べる^QFコマンドは、一般的なエディタではただちに検索を実行しますが、VZでは検索文字列を登録するだけで、検索は ^R ^C で実行します。すなわち、文字列の検索もページングの一部と考えているわけです。
検索が終了したら ^@ でページングモードを元に戻してください。

スムーススクロール

スムーススクロールは、PC-9801のもつハードウェアを生かして、ドット単位でテキスト画面をなめらかにスクロールする機能です。残念ながらPC-9801のノーマルモード以外ではこのハードウェアをもちませんが、その場合でも行単位の連続スクロール（速度可変）を行ないます。

^QW [ROLL DOWN]　:スムースアップ
^QZ [ROLL UP]　:スムースダウン

コマンドを実行したキーを押し続ける時間により、センスモードとラッチモードの2つ

のモードを使い分けることができます。(PC-9801版以外ではラッチモードのみです)

■　センスモード

実行キーを一定時間（約0.5秒）以上押し続けると、センスモードとなります。この時は、実行したキーを離すとスクロールが止まります。

■　ラッチモード

実行キーを即座に離した場合、そのままスクロールが続き、ラッチモードとなります。この時、[0]～[4]のキーでスクロール速度を変えることができます。[0]が最も低速、[4]が最も高速です（デフォルトは3です)。[0]～[4]以外のキーを押すと、スクロールは止まります。

◎　スクロール中に[SHIFT]キーを押すと、押している間、最高速（最高速の時には速度3）でスクロールします。
◎　2つのテキストを縦分割で表示している場合は、両画面スクロールとなります。この場合は、片方のテキストが先頭か最後に達した時点でスクロールは止まり、それ以上スムーススクロールしません。
◎　スムーススクロールしている間は、画面の表示色は1色になります。
◎　次のオプションで、スクロール速度の初期値を設定できます。

【*RS*3　:スムーススクロール速度（0..4）】

◎　処理速度の遅い機種では、スクロールダウン時に画面が乱れることがあります。

マークとジャンプ

指定行番号ジャンプ、タグジャンプの他、カーソル位置を4箇所までマークすることができます。

---

^QJ [SHIFT]+[f・2] J　:指定行番号ジャンプ

行番号を指定して、そこへジャンプします。コマンドを実行すると、

【行番号】

というウィンドウが開きますので、ジャンプしたい行番号を10進数で入力してください。行番号は現在の行番号モードにしたがって、論理行番号または表示行番号を入力します。

●●　行番号モード　●●

改行コードから改行コードまでの1行を「論理行」、画面の1行を「表示行」と呼ぶことは前にも触れました。ステータスラインに表示されているカーソル位置の行番号や^QJで指定する行番号は、この2つのモードに設定できます。これは、次のメニューまたはオプションで指定します。

[SHIFT]+[f・1]『N 行番号モード』

【*Dn*-　:論理／表示行番号】

表示行番号モードにすると、行数のカウントに時間を要するため、ジャンプの処理が論理行番号の時より遅くなります。

---

^QP　:直前のカーソル位置

直前のカーソル位置へカーソルを移動します。次の7つのコマンドを実行するとき、実行する直前のカーソル位置が自動的に記憶されます。

^QR	:ファイルの先頭
^QC	:ファイルの最後
^QP	:直前のカーソル位置
^QM	:カーソル位置マーク
^QJ	:指定行番号ジャンプ
^QF	:検索文字列の設定
^Q1~4	:マーク位置へジャンプ

---

^QM　:カーソル位置マーク

カーソル位置をマークします。コマンドを実行すると、

《マークしました》

と表示し、カーソル位置をマーク＃0に記憶します。^QPでマーク位置に戻ることができます。

---

^K1‾4　　　　　　　　　　　　:マーク

---

^K に続いて[1]～[4]の数字キーを入力することにより、カーソル位置をマーク＃1～＃4に記憶します。^K1 を実行すると、

《マーク #1》

と表示します。テキストごとに4箇所までマークすることができます。(^QMを含めると5箇所です)
★注意★
Ver1.5では、テキストの先頭行はマークできません。

---

^Q1～4　　　　　　　　　　　　:マーク位置へジャンプ

---

^Q に続いて[1]～[4]の数字キーを入力することにより、マーク＃1～＃4へカーソルを移動します。

[SHIFT]+[f･10]（ﾀｸﾞ）　:タグジャンプ（マクロ）

カーソル行以降のテキストでタグ行をサーチし、見つかった場合は、タグの記述に従って指定ファイルをオープンし、指定行へジャンプします。

---

^KG　　　　　　　　　　:タグ情報の取得

カーソル行を解析し、ファイル名、行番号が含まれる「タグ行」と判断した場合は、
〈ファイル名〉-〈行番号〉
の型式に変換してファイル名バッファに格納します。タグ行として参照できる書式は、次の通りです。

［書式］

［〈ヘッダー〉］〈パス名〉［〈行番号〉］：〈コメント〉

〈ヘッダー〉　　　　：．￥を含まない任意の文字列
〈パス名〉　　　　　：．￥を含む文字列
〈行番号〉　　　　　：10進数

［例］

```
MAIN.ASM(9) : Error 20: Invalid Opcode
Error tst.c 1: Unable to open include file 'stdio.h'
```

4.2 基本編集コマンド

この節では、テキストを文字・単語単位で削除・挿入する編集コマンドについて説明します。

---

^V [INS]　　　　　　:挿入モード

挿入モード (Insert) と上書モード (Overwrite) を切り替えます。上書モード時にはカーソルの形状がアンダーラインになります。それぞれのモードのカーソルの形状は、次のオプションで設定できます。

【*Ci*4　　:カーソル形状 (挿入)】
【*Co*7　　:　　〃　(上書)】

---

^H [BS]　　　　　　:バックスペース

カーソルの直前の1文字を削除し、以降の文字を1文字左へ詰めます。

---

^G [DEL]　　　　　　:1文字削除

カーソル位置の1文字を削除し、以降の文字を1文字左へ詰めます。

---

^QH [SHIFT]+[BS]　　:語の先頭まで削除

カーソルの直前の単語を削除し、以降の文字を左へ詰めます。単語の区切りについては、次節の「単語の区切り」を参照してください。

---

^T [SHIFT]+[DEL]　　:語の終りまで削除

カーソル位置の単語を削除し、以降の文字を左へ詰めます。

---

^QT ［CTRL］+［BS］　　:行頭まで削除

行頭からカーソルの直前までを削除し、以降の文字を左へ詰めます。

---

^QY ［CTRL］+［DEL］　　:行末まで削除

カーソル位置から行末までを削除します。

---

^U　　:削除文字（列）復活

削除した文字・文字列を復活し、カーソル位置に挿入します。このコマンドは、編集している行から離れても有効です。削除した文字・文字列は削除文字列バッファに順次格納され、一杯になると、古いものから捨てられています。このバッファのサイズは、次のオプションで設定できます。

【 ***Bc***256　:削除文字列バッファサイズ 】

［TAB］　　:タブ

挿入モードと上書きモードで動作が異なります。

■ 挿入モード

タブコード（09h）を挿入します。「オプション***Eu***」を「+」に設定すると、タブの大きさと同じ数の半角スペースを挿入します。

【 ***Eu***-　:［TAB］でスペースコードを入力 】

■ 上書モード

カーソルを次のタブ位置へ移動します。

タブサイズの標準値は8（拡張子が .c .h のファイルは4）です。この値は、次のオプションで1, 2, 4, 8, 16, 32, 64に設定できます。

【 ***Ta***8　:タブカラム数 】

【 ***Tb***4　:.C .H のタブカラム数 】

また、編集中に次のメニューで8または4に変更できます。

[SHIFT]+[f・2] 『T タブの切替え』

[↵] :リターン

挿入モードと上書きモードで動作が異なります。

■ 挿入モード時

カーソル位置に改行コード (0Dh 0Ah) を挿入します。オートインデントがONの場合は、カーソルの前に上の行のスペース・タブが複写されます。オートインデントモードは次のメニューまたはオプションでON／OFFできます。

[SHIFT]+[f・1] 『A オートインデント』

【 *Ei*+ :オートインデント 】

■ 上書モード時

カーソルを次の表示行の先頭に移動します。ファイルの終りの行で実行した場合は、挿入モードと同じ動作となります。

---

^N [SHIFT]+[↵] :1行挿入

カーソル行（論理行）の前に、改行を挿入します。オートインデントがONの場合は、カーソルの前に下の行のスペース・タブが複写されます。

---

^QU :1語大／小文字変換

カーソル位置の文字が半角英大（小）文字ならば、以降の1単語中の英字を小（大）文字に変換します。

[例]

```
           ^QU              ^QU
PopUpMenu  →  POPUPMENU  →  Popupmenu
```

---

^QN :ファイル名の複写

現在編集中のテキストのフルパス名を、カーソル位置に挿入します。編集中のテキストをコンパイルするようなマクロの中で使用します。

●●　単語の区切り　●●

単語の区切りは、次に示す文字のタイプに基づいて動作します。

タイプ2	［半角］スペース、タブ、制御文字（00h～20h） ［全角］スペース（8140h）
タイプ3	［半角］'$'　'_'を除く英記号 ［全角］記号（8152h～815Bhを除く）
タイプ4	全角ひらがな
タイプ5	上記以外の全角文字
タイプ6	上記以外の半角文字

これらのタイプ番号の大小を、カーソル位置とその直前の文字で比較します。

■ ^A ［SHIFT］+［←］コマンド
《直前の文字タイプ》＜《現在の文字タイプ》となる位置まで左へ移動します。

■ ^F ［SHIFT］+［→］コマンド
《直前の文字タイプ》＞《現在の文字タイプ》となる位置まで右へ移動します。

■ ^QH ^T ［SHIFT］+［BS］［SHIFT］+［DEL］コマンド
《直前の文字タイプ》と《現在の文字タイプ》が異なる位置まで削除します。

以上のように、英単語以外に日本語の文章でも漢字／ひらがなの境界を利用して、単語単位の処理が可能です。

4.3 ヒストリー機能

これまで説明してきたカーソルの移動・編集コマンドのうち、1行内の操作に関するコマンドは、ファイル名、検索文字列、コマンドライン等の文字列入力モード時にも使用できます。これらのコマンドは、次のとおりです。

^S	[←]	:←
^D	[→]	:→
^A	[SHIFT]+[←]	:1語←
^F	[SHIFT]+[→]	:1語→
^QS	[CTRL]+[←]	:論理行頭
^QD	[CTRL]+[→]	:論理行末
^V	[INS]	:挿入モード
^H	[BS]	:バックスペース
^G	[DEL]	:1文字削除
^QH	[SHIFT]+[BS]	:語の先頭まで削除
^T	[SHIFT]+[DEL]	:語の終りまで削除
^QT	[CTRL]+[BS]	:行頭まで削除
^QY	[CTRL]+[DEL]	:行末まで削除
^U		:削除文字（列）復活
^I	[TAB]	:タブ
^P		:制御文字入力
^QU		:1語大／小文字変換
^QI	[SHIFT]+[f・5]	:文字列の複写
^QN		:ファイル名の複写

また、文字列入力時には、「ヒストリー機能」が使用できます。これは、入力した文字列を「ヒストリーバッファ」と呼ぶ内部バッファに記憶し、入力行に呼び出して使用する機能です。この機能を用いると、同じ文字列を何度も入力する場合や、以前に入力した文字列の一部を変更して入力する場合等に、入力の手間が大幅に省けます。

ヒストリーバッファの文字列は、次のコマンドで呼び出します。

---

^E [⬆]　　　　　　　　:ヒストリー↑

ヒストリーバッファの、より古い文字列を順次表示します。最も古い文字列を表示している場合は何も起こりません。

---

^X [⬇]　　　　　　　　:ヒストリー↓

ヒストリーバッファの、より新しい文字列を順次表示します。最も新しい文字列を表示した後は、入力行がクリアされます。

また、

　　str[⬆]

のように、文字列の先頭の数文字を入力した後に[⬆]を実行すると、「str」で始まる文字列のみ順次サーチすることができます。その後の[⬇]で逆方向にサーチします。「str」で始まる文字列が存在しない場合は何も起こりません。「str」のサーチをキャンセルする場合は、[⬅][⬇]を押して、入力行をクリアしてください。

■　ヒストリーのワード検索

さらに、Ver1.5では、ワード単位の検索が可能になりました。次のように、入力行のカーソル位置以前にスペースが入っている場合は、ワード単位のヒストリー検索を行います。(ここでいう「ワード」は、スペースで区切られた文字列です)

[例1]

　　【入力ファイル】

　　aaa bbb ccc ddd[↵]

　　【入力ファイル】

　　aaa c[⬆]

　　aaa ccc [⬆]

　　aaa ccc ddd

[例2]

```
A>copy aaa b:[↵]
...
A>dir bbb[↵]
...
A>type [⬆]
```

```
A>type bbb[⬆]
A>type aaa[⬆]
A>type b:
```

[例2] のように、コマンドラインの場合は、先頭の1ワードはコマンドとみなしてスキップします。

■ ヒストリーバッファの種類

ヒストリーバッファ名	サイズ指定オプション	ウィンドウのタイトル
文字列	Hs256	検索文字列　置換文字列　複写文字列
ファイル名	Hf128	入力ファイル　新規ファイル　追加ファイル
コマンドライン	Hx128	
アプリケーション	Ha64	
ワーク	Hw64	行番号　マクロ文

■ ヒストリーの学習機能

ヒストリーバッファでは、常に最後に入力（またはヒストリー機能で選択）したものを、最初に表示するように、バッファ内で並び換えを行なっています。また、入力した文字列がバッファ内に記憶している文字列と一致した場合には、古い文字列を削除し、最後に入力したものを先頭に記憶します。

また、ヒストリーバッファが一杯になった場合は、古い文字列から消去します。

■ ヒストリーバッファの初期文字列

各々のヒストリーバッファには、DEFファイルに初期文字列を格納しておくことができます。

```
* H ヒストリー

:string
          include define static extern ...
:file
          "vz.def vz.doc"
```

ヒストリーバッファのタイプは、行の先頭の「:<英字>」で識別し、その間に初期文字列をスペース、または改行で区切って記述します。文字列中にスペースを指定したい場合は、文字列を「"」でくくってください。タイプの指定は次のとおりです。

:s...　　文字列
:f...　　ファイル名
:c...　　コマンドライン
:a...　　アプリケーション
:w...　　ワーク

大文字／小文字は区別せず、先頭の1文字しか識別しません。

初期文字列を記述すると、ヒストリーバッファのサイズはその分大きくなります。

4.4 カット＆ペースト

VZのカット＆ペースト処理は、次のような従来のエディタにはない特徴をもっています。

- 「ブロックモード」による、行およびブロックの共通操作
- 「テキストスタック」による、複数の行およびブロックの格納
- カーソルの移動方向による、行単位と文字単位のブロック境界切り替え

また、矩形ブロックの操作もマクロで実現しました。
カット＆ペーストの操作コマンドは、ファンクションキーの右側に割り当てられています。

<ファンクションキー>

	[ﾌｧｲﾙ]	[窓換]	[文換]	[窓割]	[記憶]	[検索]	[置換]	[ｶｯﾄ]	[ｲﾝｻｰﾄ]	[ﾌﾞﾛｯｸ]
ｼﾌﾄ	[ﾓｰﾄﾞ]	[ｻﾌﾞ]	[比較]	[窓動]	[複写]	[ﾏﾙﾁ]	[複写2]	[ｺﾋﾟｰ]	[ﾍﾟｰｽﾄ]	[ﾀｸﾞ]

^B	[f･10]	:ブロックモード
^B ^KB		(J-3100, IBM PC)

カーソル位置をブロックの境界とし、ブロックモードになります。（この時、画面最上部のステータスラインの □ マークが反転します）そこから、カーソルを移動した位置までをブロック領域として、反転表示します。再度 ^B を実行すると、ブロックはキャンセルします。
以下のコマンドは、通常行単位またはテキスト全体で処理しますが、ブロックモードで実行した場合は、ブロック領域を対象とします。

^Y	[f･8]	:行･ブロック削除
^KK	[SHIFT]+[f･8]	:行･ブロック記憶
^QA	[f･7]	:文字列の置換
^L	[f･5]	:検索文字列の取得
^KH		:インデントの変更
[ESC] S		:ファイルのセーブ
[ESC] A		:既存ファイルへ追加

ブロックの境界は、^B 実行後、カーソルを上下に移動している間は「論理行単位」で、左右に移動したり編集操作を行なうと「文字単位」になります。この場合は、再び上下に移動すると「論理行単位」に戻ります。また、ブロックが論理行内におさまっている場合は、常に文字単位の境界となります。

ブロック領域の反転表示は、行単位では画面の右端まで、文字単位では行末までとなりますから、一目で境界のモードを知ることができます。

---

^Y [f・8] (ｶｯﾄ)　　　:行・ブロック削除

カーソル位置の論理1行、またはブロックをテキストから削除し、テキストスタックへ格納します。

---

^J [f・9] (ｲﾝｻｰﾄ)　　　:行・ブロック挿入

テキストスタックに最後に格納した行またはブロックをカーソル位置に挿入し、スタックから削除します。行や行単位のブロックは、カーソル行（論理行）の前に挿入します。また、「オプション ***Ek***」を「+」に指定すると、ブロックを挿入した時、カーソルは挿入したブロックの最後へ移動します。

【 ***Ek***-　　:ブロックペースト後カーソル位置更新 】

---

^KK ［SHIFT］+［f・8］(ｺﾋﾟｰ)　:行・ブロック記憶

カーソル位置の論理1行、またはブロックを、テキストスタックへ格納します。テキストはそのままです。

---

^KC ［SHIFT］+［f・9］(ﾍﾟｰｽﾄ)　:行・ブロック複写

テキストスタックに最後に格納した行やブロックを、カーソル位置に挿入します。スタックからは削除しません。行や行単位のブロックは、カーソル行（論理行）の前に挿入します。^J と同様に挿入後のカーソル位置を「オプション ***Ek***」で制御できます。

---

^KU　:ペーストのアンドゥ（マクロ）

直前に ^J ^KC コマンドでペーストしたブロックを、ふたたび削除してテキストスタックに格納します。

---

^KY ［SHIFT］+［f・2］C　:スタックの消去

テキストスタックをクリアします。コマンドを実行すると、

《テキストスタックを消去しますか？（Y/N）》

と聞いてきますので、以下のキーで答えてください。

［Y］　：テキストスタックを消去
［N］,［ESC］　：コマンドの実行を中止

ただし、標準設定ではテキストスタックが一杯になった場合は、自動的に古いものから捨てられますので、通常このコマンドを使用する必要はありません。次のオプションを指定すると、テキストスタックが一杯になった時に、上の消去確認メッセージを出します。

【 ***Ex***-　:テキストスタックfull時の警告 】

テキストスタックのサイズは、次のメニューでステータスラインのインジケータの部

分にシステム情報を表示すると、そこに「xxx/stack」で示します。

　　[SHIFT]+[f･1]『S システム情報の表示』

---

^QB [SHIFT]+[f･2]B　:ブロックの先頭／最後

■ ブロックモード

カーソル位置をブロックの始点とし、もとのブロックの始点へジャンプします。すなわち、ブロックの先頭／最後を交互に参照できます。

■ 非ブロックモード

最後に^B ^QB コマンドを実行した位置へジャンプします。

■ ^J ^KCコマンド実行直後

挿入したブロックの最後（*Ek*+の時は先頭）へカーソルを移動します。

●● 大きなブロックの削除 ●●

テキストスタックは、メモリが十分ある状態で、最大64KBまで確保できます。しかし、64KBを越えるようなブロックは、一度に格納することはできません。このような大きなブロックに対して ^Y ^KK コマンドを実行すると、

　　《ブロックが大きすぎます》

と表示し、64KBに収る範囲のみをブロックに格納し、残りの部分はブロックとして残します。従って、このメッセージが出なくなるまで、続けてコマンドを実行することで、大きなブロックも削除することができます。

矩形ブロック操作

矩形ブロック操作は、矩形ブロックマクロを記述した「block.def」ファイルを組み込まないと、使用できません。以下のいずれかの方法で組み込んでください。

(1) 起動時にコマンドラインで指定

A>vz +block[↵]

(2) DEF ファイル内で指定

* E その他

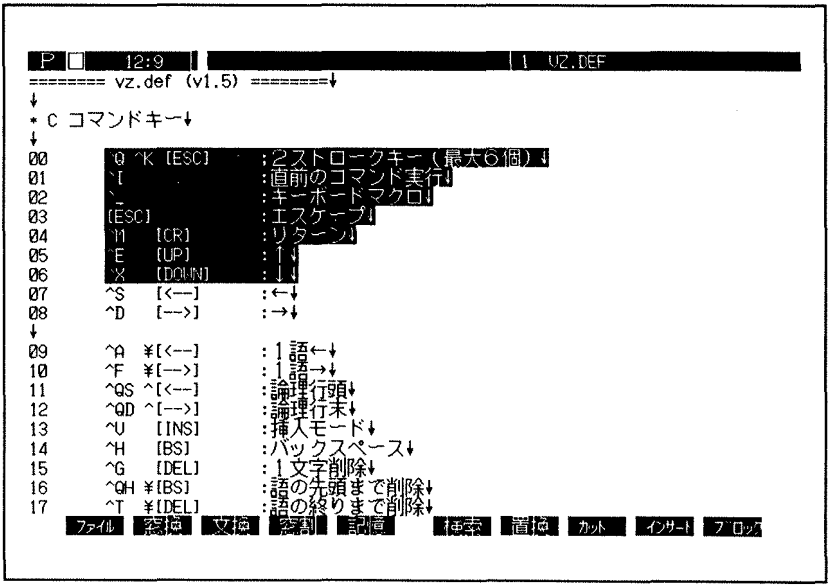
1　　　block

(3) 編集中に[**ESC**]^で外部マクロとして実行

矩形ブロックマクロを組み込むと、次のコマンドで矩形ブロックモードになります。

^KB　　　:矩形ブロックモード

^KBを実行したら、カーソルをそのまま下へ移動してみてください。カーソル位置の右側の部分が反転表示されます。



今度は、カーソルを右へ移動します。すると、矩形ブロックの始点とカーソル位置で囲まれた矩形エリアが反転します。

```
 P       12:26                                        1  VZ.DEF
======== vz.def (v1.5) ========↓
↓
* C コマンドキー↓
↓
00      ^Q ^K [ESC]     :2ストロークキー（最大6個）↓
01      ^[              :直前のコマンド実行↓
02      ^_              :キーボードマクロ↓
03      [ESC]           :エスケープ↓
04      ^M   [CR]       :リターン↓
05      ^E   [UP]       :↑↓
06      ^X   [DOWN]     :↓↓
07      ^S   [<--]      :←↓
08      ^D   [-->]      :→↓
↓
09      ^A  ¥[<--]      :1語←↓
10      ^F  ¥[-->]      :1語→↓
11      ^QS ^[<--]      :論理行頭↓
12      ^QD ^[-->]      :論理行末↓
13      ^U   [INS]      :挿入モード↓
14      ^H   [BS]       :バックスペース↓
15      ^G   [DEL]      :1文字削除↓
16      ^QH ¥[BS]       :語の先頭まで削除↓
17      ^T  ¥[DEL]      :語の終りまで削除↓
   ファイル 窓換 文換 窓割 記憶   検索 置換 カット インサート ブロック
```

このように、ブロックの始点とカーソル位置のカラム位置の関係で、矩形エリア、または各行の行末までを選択できます。

矩形エリアを指定したら、通常のブロック操作コマンドで、カット&ペーストを実行できます。ただし、^**KK** のコピーは使用できません。すなわち矩形ブロックは、一度ペーストすると失われてしまいます。

行末以降でペーストした場合は、行末の短い部分に、半角のスペースまたはタブを補います。タブを使いたくない場合は、[**TAB**]のところで触れた「オプション***Eu***」を「+」に設定してください。

矩形ブロックマクロを組み込むと、次のコマンドも使用できます。

[**ESC**]]　　　　　　　　:行末のSPACE・TABの削除

指定したブロック内の行末のスペース・タブを取り去ります。ブロックモードでない時は何も実行しません。このコマンドは、例えばDEFファイルのコメントの部分を矩形ブロックとして削除した後、その間の余計な行末のスペース・タブを取り去る時に使用します。

4.5 文字列の操作

この節では、文字列の検索・置換・複写操作を説明します。いずれのコマンドも、入力した文字列は「検索文字列バッファ」に保存し、再度使用する時にはヒストリー機能により簡単に呼び出すことができます。
文字列の操作コマンドは、ファンクションキーの中央に割り当てられています。

＜ファンクションキー＞

	[ﾌｧｲﾙ]	[窓換]	[文換]	[窓割]	[記憶]	[検索]	[置換]	[ｶｯﾄ]	[ｲﾝｻｰﾄ]	[ﾌﾞﾛｯｸ]
ｼﾌﾄ	[ﾓｰﾄﾞ]	[ｻﾌﾞ]	[比較]	[窓動]	[複写]	[ﾏﾙﾁ]	[複写2]	[ｺﾋﾟｰ]	[ﾍﾟｰｽﾄ]	[ﾀｸﾞ]

---

^QF [f・6]（検索）　　:検索文字列の設定

検索文字列を設定します。このコマンドは、文字列を設定するだけで、検索はしません。検索は ^R ^C で行ないます。コマンドを実行すると、

【検索文字列】

というウィンドウが開きますので、検索する文字列を入力してください。[↵]で入力を終えると、最下行に黄色で検索する文字列を表示し、ページングモードが

S, St（文字列検索モード）

に切り替わます。（ステータスライン左端にモードを表示）
文字列検索モードは、^R ^C で、ページングの代わりにカーソル位置の前方または後方へ文字列を検索し、見つかった場合は、その文字列の先頭へカーソルを移動します。指定の方向に文字列が見つからない場合には、

《文字列が見つかりません》

と表示され、カーソルは移動しません。

検索を終了したら、^@ コマンドでページングモードを元に戻してください。検索文字

列設定後は、ページングモードはP→C→Sの3モードで切り替わります。検索文字列をキャンセルする場合は、^QF コマンドで、[ESC]、または [↵]のみを入力してください。このときページングモードは、P→Cの2モードになります。

---

^QA [f・7]（置換）　　:文字列の置換

文字列の置換を行ないます。指定ブロック内の文字列を置換する場合は、あらかじめブロックを指定しておきます。コマンドを実行すると、

【検索文字列】

というウィンドウが開きますので、検索する文字列を入力してください。続いて、

【置換文字列】

というウィンドウが開きますので、今度は置換したい文字列を入力してください。ここで、ブロックモードでない場合には、

置換範囲
A テキスト全体
F カーソル以降
B カーソル以前

というメニューが表示されますので、置換を実行する範囲を選択します。最後に、

《全部一度に置換しますか？ (Y/N)》

と聞いてきますので、以下のキーで答えてください。

[Y]　　:確認なしで指定範囲の置換を実行
[N]　　:確認付きで置換を開始

[ESC]　　　　　　：コマンドの実行を中止

確認付きの置換の場合は、検索した文字列へカーソルを移動し、

《置換しますか？ (Y/N)》

と聞いてきますので、以下のキーで置換するかどうか指定します。

[Y]　　　　　　：置換
[N]　　　　　　：置換しせず
[ESC]　　　　　：コマンドの実行を中止

全ての置換が終了すると、

《xx個の文字列を置換しました》

と表示し、コマンドを終了します。

※　文字列の検索、または置換を途中で中断したい場合は、[STOP]キー（J-3100版、IBM　PC版では[Ctrl]+[Break]キー）を押してください。

---

^QO　　　　　　　:直前の置換の再実行

直前に実行した設定の置換（検索・置換文字列、置換範囲、確認の有無）を、再実行します。

[SHIFT]+[f･6]（ﾏﾙﾁ）　:複数テキストでの検索･置換（マクロ）

直前に実行した検索または置換を、オープンしている全てのファイルに対して実行します。

---

^QK [SHIFT]+[CLR]　　:対応する() [] {} <>
　[Shift]+[Home]　　(J-3100, IBM PC)

カーソル位置の文字が半角の「( ) [ ] { } < >」の場合、対応する括弧を検索し、カーソルを移動します。カーソル位置の文字が上記の括弧でない場合、あるいは対応する括弧が見つからない場合には、

《括弧が見つかりません》

と表示します。ただし括弧の上にカーソルがない場合でも、編集テキストの拡張子が「.c .h」の場合は、「}」の上にあるものとして、対応する「{」を検索します。従って、カーソルをブロックの始点へジャンプできます。

---

^L [f•5]（記憶）　　　　:検索文字列の取得

検索文字列をカーソル位置のテキストから取得します。コマンド実行後は、^QFと同様に文字列検索モードになります。ブロックモードか否かによって処理が違います。

■ 非ブロックモード

カーソル位置からの1単語を、検索文字列として取得します。Ver1.5付属のDEFファイルでは、連続取得のマクロが組み込まれており、その場合には^L を連続して押すことで、順次次の単語境界までの文字列を、取得することができます。

■ ブロックモード

ブロック指定した文字列を検索文字列として取得します。ただし、ブロックが複数行にまたがっている場合は、何も実行しません。

---

^QI [SHIFT]+[f•5]（複写）　:文字列の複写

直前に^QF ^L コマンドで定義した検索文字列を、カーソル位置に挿入します。

---

^KA [SHIFT]+[f•7]（複写2）:文字列の選択複写

コマンドを実行すると、

```
┌【複写文字列】─────────────┐
│String                          │
└────────────────────────┘
```

というウィンドウが開き、直前に設定した文字列を表示します。ここで、ヒストリー機能を使ってカーソル位置に挿入したい文字列を選択してください。

[CTRL]+[TAB]	:文字列の補完（マクロ）
[Shift]+[Tab]	(J-3100, IBM PC)

ある文字列の最初の数文字を入力した場合、その文字で始まる文字列を検索文字列バッファで探して、見つかった場合は残りの部分を補完します。コマンドを続けて実行すると、見つかった文字列を順番に挿入します。

[例]

"int" "include" が、この順でバッファに格納されている場合、

i□ → int□ → include□ → int□

#inc□ → #include□

よく使う予約語をDEF ファイルの

```
* H ヒストリー

:string
        include define static extern ...
```

のところに登録しておけば、このように短縮入力ができます。

検索文字列の特殊文字

*	ワイルドカード。文字列の両端で指定できます。
¥n	0Dh+0Ah　(改行コード)
¥r	0Dh
¥l	0Ah
¥*	* 文字
¥¥	¥ 文字

★注意1★
「¥n」等の改行コードを、^QA コマンドで他の文字に置換した場合、行番号がずれる場合があります。その場合はいったん ^QR コマンドでテキストの先頭にジャンプし、^QP コマンドで元の位置に戻ってください。
★注意2★
テンポラリファイルを作成する大きなファイルで、「¥n」を使って行頭の文字列を置換すると、置換できない文字がいくつか発生します。

■　ワードサーチ

ワードサーチとは、指定した文字列が、テキスト中で両端が
- スペース、タブ、改行、制御文字（00h～20h）
- '$' '_' を除く半角英記号
- 全角文字

で区切られている場合に、一致するとみなして検索するモードです。ワードサーチがOFFの場合は、検索文字列が文字列の一部に含まれれば、一致していると見なします。
[例]
「i」を次のテキストでワードサーチした場合、「^」の付いた「i」だけを検索します。

```
for (i=0; i<n; i++) {
     ^    ^    ^
    if( list(i) ) break;
             ^
}
```

これに対して、ワードサーチがOFFの時は、全ての「i」の文字を検索します。
ワードサーチの時、検索文字列の前または後に「*」を付けると、その側の文字が他の文字列とつながっていても、一致するとみなします。例えば上の例で「i*」を指定すると「if」を、「*i*」を指定すると「if」と「list」も一致していると解釈します。

ワードサーチモードがOFFの場合に検索文字列に「*」を付けると、今度は逆にその側のワード境界をチェックします。

ワードサーチモードは、次のメニューまたはオプションで変更できます。

[SHIFT]+[f・1]　『 W ワードサーチ 』

【 *Ew+*　　:ワードサーチ 】

ただし、^QF, ^L コマンドで検索文字列を設定した後で切り替えても、その文字列の検索モードは変わりません。

ワードサーチは、半角英単語（英数字）の検索に限ります。すなわち、検索文字列の端が、

・英数字、'$'、'_' 以外の半角文字

・全角文字

の場合は、その側はワード境界のチェックはしません。従って日本語の文章で全角文字列を検索する際は、ワードサーチは無効になります。

■　英大／小文字の識別

検索の際に、半角英大文字と小文字の識別をするかしないかを、次のメニューまたはオプションで指定できます。

[SHIFT]+[f・1]　『 I 検索時の英大/小文字 』

【 *Ec-*　　:検索時の英大／小文字の同一視 】

4.6 その他のコマンド

---

^[　　　　　　　　　　　:直前のコマンド実行

直前に実行したコマンドまたはマクロを再実行します。2ストロークコマンドや、オートリピートの効かないキーに割当てられたコマンドを、連続して実行させたい場合に使用すると便利です。

---

^QL　　　　　　　　　:行の回復

カーソルがある論理行に対して行なった編集をキャンセルし、編集を行なう前の行を表示します。

[**CLR**]　　　　　　　　:表示位置変更（マクロ）
[**Home**]　　　　　　　　(J-3100, IBM PC)

画面上でのカーソルのY位置を、

最上行－1 → 中央 → 最下行－1

と変更します。テキストもカーソルと一緒に動きますから、テキスト上でのカーソルの位置は、変りません。

---

^KI [SHIFT]+[f・2] T　:タブの切り替え（マクロ）

編集中のテキストのタブサイズを、8または4に切り替えます。

---

^KL [SHIFT]+[f･2] L　:表示行数切り替え

画面表示行数を機種によって次のように切り替えます。

PC-9801（ノーマル）	25／20
PC-9801（ハイレゾ）	25／31
J-3100	変更できません
IBM PC（CGA）	変更できません
IBM PC（EGA）	25／43
IBM PC（VGA）	25／50

---

^KH [SHIFT]+[f･2] I　:インデントの変更

カーソル行やブロックのインデント（字下げ）を変更します。コマンドを実行すると、

《カーソルキーで位置を移動して[↵]》

と表示されますので、以下のキーでインデントを変更してください。

^S [←] ･････････････	1カラム分減らす
^D [→] ･････････････	1カラム分増やす
^A [SHIFT]+[←] ･････	タブカラム数分減らす
^F [SHIFT]+[→] ･････	タブカラム数分増やす
[↵] ････････････････	確定する

行頭のインデントの部分には、可能な限りタブコードを挿入します。

★注意★

インデントを増減した時点で、テキストを書き換えますので、キャンセルはできません。

---

^KX [SHIFT]+[f･3]（比較）　:テキストの比較

2つ以上のテキストをオープンしている時、表示中の2つのテキスト（全画面表示の場

合は裏テキスト）を比較します。コマンドを実行すると、カーソル位置からテキストを比較し、相違があった場合にはその位置へカーソルを移動し、

《相違がありました》

と表示します。テキストの最後まで一致した場合には、

《一致しました》

と表示し、カーソル位置を変えずにコマンドの実行を終了します。

---

^KJ　　　　　　　　　　:文字コード表示（マクロ）

カーソル位置の文字コードを、画面最下行に表示します。[**ESC**]で終了します。

---

^KP　　　　　　　　　　:プリント専科の起動（マクロ）

編集中のテキストをセーブして、「ps <ﾌｧｲﾙ名>」の形式でコマンドを実行します。標準設定では「プリント専科Ⅱ」を呼び出しますが、DEF ファイルの「ps」の箇所を変更すれば、他のユーティリティも起動できます。

4.7 特殊文字の入力

---

^P　　　　　　　　　:制御文字入力

制御文字 (00h～1Fh) を入力します。コマンドを実行すると、ファンクションキー表示の左端に、「^」が表示されますので、続いて[英字]または ^[英字]キーで入力してください。

[例]

^PG　　　: ベル文字 (07h)
^PL　　　: 改ページ文字 (0Ch)
^P[　　　: エスケープ文字 (1Bh)

※ **^PJ** でLFコード (0Ah) を入力することはできません。

入力した制御文字は、編集テキスト上では水色の「^」+英字で表示します。

★注意★

^P 押下時にプリンタの電源が入っていて、オフラインになっている場合、プリンタにLFコードを送ろうとしてハングアップしてしまう場合があります。この場合は、一度プリンタの電源を切ってください。

[CTRL]+[テンキー]　　　(PC-9801版のみ)
[GRPH]+[テンキー]　　　(ezkey.com常駐時)

罫線・記号文字を入力します。どのような文字を入力するかは、DEFファイルの以下の箇所に指定できます。

```
* E その他
...
5        0●,◎.○1 《2↓3》4←5■6→7 『8↑9』=〃+※*★- 【/】
6        0└,┴.┘1 ├2┼3┤4 ├5┼6┤7 ┌8┬9┐=|+—*—/———
```

5番で[**GRPH**]+[**テンキー**]、6番で[**CTRL**]+[**テンキー**]を指定します。この機能を使用しない場合は、それぞれの行を削除してください。

第5章　ファイラー

5.1 ファイラーの概要
5.2 ファイラーの基本操作
5.3 メニューコマンド
5.4 複数ファイルのコピー

5.1 ファイラーの概要

前章まで、VZのエディタとしての機能について説明してきました。この章からは編集作業をとりまく様々な操作に対するVZの機能について説明します。

エディタでテキストを編集する場合、まず初めに編集テキストを指定しなければなりません。これはファイル名を打ち込んでもよいのですが、すでにあるファイルを編集する時は、ディレクトリを表示して、そこからファイルを選択した方がはるかに楽です。VZでは、このファイル選択モードをさらに拡張し、「ファイラー」と呼ばれる強力なツールを構成しました。これは次のような機能をもっています。

- ・サブディレクトリー間の移動
- ・ロング／ショートの表示モード
- ・ディレクトリのソート表示
- ・複数の拡張子による選択
- ・2つのディレクトリを同時表示
- ・2つのディレクトリの比較
- ・選択したファイルに対するコマンドの連続実行

VZのファイラー自体は、ファイルのコピーや削除の機能はもっていません。これらの機能は、コマンドとして「command. com」にまかせています。

ファイラーは、次のタイミングで起動します。

- ・『入力ファイル』ウィンドウで、パス名やワイルドカードを指定した場合。
- ・最後の編集テキストをクローズし、前回ファイラーからオープンしている場合。
- ・コマンドラインモードで［SHIFT］+［↵］を実行した場合。

コマンドラインからファイラーを起動した場合は、コマンドの実行と密接な関わりをもってきますが、これについては『6.3 ファイラーとの連携』で説明します。

ファイラーの画面は、次のようになっています。

```
Drive  Mask  Path  Sort  View  Window  All  TrCD  RetCD  Compare  Exec ←①
┌ A:¥VZ¥*.* ←② ─────────────────┐
│                               │
│          ③                    │
│                               │
│                               │
└── xx files ── xxxxx bytes free ──┘
       ↑            ↑     (total)
       ④            ⑤
```

① メニューコマンド（先頭の英字1文字でコマンドを実行できます）
② ディレクトリ名（カレントディレクトリの場合は、反転表示）
③ ディレクトリ内容
④ 表示または選択（反転）しているファイルの総数
⑤ ディスクの空き領域サイズ、または選択したファイルが消費する全クラスタサイズ
　・1ディレクトリ表示の場合 …… ソースドライブのクラスタ単位
　・2ディレクトリ表示の場合 …… ターゲットドライブのクラスタ単位

※ ④、⑤は、選択したファイルがない時とある時で表示する内容が変ります

5.2 ファイラーの基本操作

まず最初に、ファイラーの基本操作である、カーソルの移動、ファイルの選択機能について説明します。カーソル位置は、「黄色のアンダーライン（PC-9801版）」で表示し、選択したファイルは反転表示します。

---

^E ^X ^S ^D [↑] [↓] [←] [→]

カーソルを移動します。2つのディレクトリを縦分割で表示している場合、横方向の移動キーでもう一方のディレクトリへ移動できます。

[SPACE]

カーソル位置のファイルを選択します。既にそのファイルが選択してある時は、選択を取り消します。選択したファイルは反転表示します。

[SHIFT]+[↓] [SHIFT]+[→]

カーソル位置のファイルを選択または非選択した後、カーソルを移動します。

[SHIFT]+[↑] [SHIFT]+[←]

カーソルを移動した後、カーソル位置のファイルを選択または非選択します。

[BS]

カーソルをディレクトリの先頭に移動します。すでに先頭にある時は、ディレクトリの最後に移動します。

[TAB]

2つのディレクトリを表示している場合に、もう一方のディレクトリへカーソルを移動します。

[↵]

■　選択したファイルがない場合
カーソル位置のファイルを選択して、ファイラーを終了します。

■　すでに1つ以上のファイルが選択してある場合
ファイラーを終了します。

■　カーソルがサブディレクトリにある場合
そのディレクトリに移動します。

ファイルを[↵]で選択すると、通常それらのファイルをオープンして編集モードになります。ただし、ファイラーを起動した状況、あるいは選択したファイルの拡張子によっては、コマンドラインへ移行する場合があります。これについては『6.3 ファイラーとの連携』で説明します。

[ESC]

選択をキャンセルし、ファイラーを起動した状態に戻ります。

★注意★　カレントドライブ以外のドライブで、サブディレクトリ間の移動を行なった場合は、最後に表示したディレクトリが、そのドライブのカレントディレクトリとなります。

Ver1.5から、以下の編集コマンドがファイラーでも使えるようになりました。

PC-9801版

^W	[SHIFT]+[ROLL DOWN]	:ロールアップ
^Z	[SHIFT]+[ROLL UP]	:ロールダウン
^R		:ページアップ
^C		:ページダウン
^QR	[CTRL]+[ROLL DOWN]	:ディレクトリの先頭
^QC	[CTRL]+[ROLL UP]	:ディレクトリの最後

J-3100,IBM PC／AX版

^W	[Shift]+[Page Up]	:ロールアップ
^Z	[Shift]+[Page Down]	:ロールダウン
^R	[Page Up]	:ページアップ
^C	[Page Up]	:ページダウン
^QR	[Ctrl]+[Page Up	:ディレクトリの先頭
^QC	[Ctrl]+[Page Down]	:ディレクトリの最後

ページアップ／ダウンは、「オプション ***Eh***」に従って全画面または半画面ページングします。

●●　ディレクトリの再読み込み　●●

Ver1.5から、ファイラーを起動して[ESC]でキャンセルし、再びファイラーを起動したような場合、ディレクトリの内容を再度読み込まなくなりました。ただし、フロッピーディスクのようにメディアの交換が可能なドライブや、ファイルをセーブしたりDOSコマンドを実行した後は、ディレクトリの内容が変更された可能性があるため、必ずディレクトリを読み込みます。

5.3 メニューコマンド

メニューコマンドは、画面最上部のメニューの先頭の英字1文字を入力することにより実行できます。

Drive　　　　　:ドライブ

ドライブを変更します。[D]を入力すると、次のメニューを表示します。

```
Drive  Mask  Path  Sort  View  Window  All  TrCD  RetCD  Compare  Exec
      .*
  A:  ──── <End of Dir> ──────
  B:   files ── 1250304 bytes free ─
  C:
  D:
  E:
```

カーソルキーまたはドライブ名に相当する英字キーでドライブを選択してください。そのドライブのディレクトリを表示します。

ドライブメニューで、実際に接続しているドライブ以上のものが表示される場合があります。これは、MS-DOS3.1から対応した「仮想ドライブ」の機能により、割り当て可能な最大ドライブ数まで表示されるためです。これを実際のドライブ数にしたい場合は、「config.sys」ファイルに

　　　LASTDRIVE = A

の1行を加えてください。

Mask　　　　　　:マスク

ファイルの拡張子を変更します。[M]を入力すると、次のメニューを表示します。

```
Drive  Mask  Path  Sort  View  Window  All  TrCD  RetCD  Compare  Exec
┌ B:¥*.  ┌────────────┐
│------  │W *.*       │ir> ----------- │
└──── 0  │Q .         │04 bytes free ──┘
         │C .h .c     │
         │A .inc .asm │
         │E .exe .com │
         │B .bat      │
         │D .doc .txt │
         └────────────┘
```

ディレクトリの中で表示したい拡張子を選択してください。この内容は、DEF ファイルの次の箇所で定義してありますので、自由に設定できます。

```
* S システムメニュー
...
5       "",13,7
        "W *.*"
        "Q ."
        "C .h .c"
        "A .inc .asm"
        "E .exe .com"
        "B .bat"
        "D .doc .txt"
```

※　メニューの個数を変える場合は、最初の行の「7」をメニューの個数に変更してください。

このメニューで、「Q .」を選択すると、デフォルト拡張子、すなわち

.doc .txt .bat .def .h .c .inc .asm

を選択します。

Path　　　　　:パス

新たに表示するパス名、拡張子、ワイルドカードを使ったマスクを指定します。[P]を入力すると、

```
Drive  Mask  Path  Sort  View  Window  All  TrCD  RetCD  Compare  Exec
┌ B:¥*.* ─────────────────────────────────┐
│ ─────────── <End of Dir> ─────────── │
└┌─ 【パス・マスク】 ──────────────────┐
  │                                        │
  └────────────────────────────────────┘
```

というウィンドウが開きますので、編集ファイル指定と同様に、パス名またはマスクを入力してください。いくつか例をあげましょう。

[例]

b:	: ドライブBの全ファイル
*.def	: 拡張子 .def のファイル
.def	:*.def と同じ
.inc .asm	: 拡張子 .inc または .asm のファイル
c:¥h¥	:c:¥h¥ の全ファイル
c:¥h	:c:¥h¥と同様

※ Ver1.5から、ディレクトリ指定の終わりの「¥」は、省略できます。

■ VZPATHのメニュー

パス・マスクの入力時に[↵]のみ入力すると、環境変数「VZPATH」で設定したディレクトリのメニューを表示します。メニューを選択すると、ただちに該当するディレクトリを表示することができます。この機能はVer1.5で追加されました。

```
Drive  Mask  Path  Sort  View  Window  All  TrCD  RetCD  Compare  Exec
┌ B:¥*.* ─────────────────────────────────┐
│ ┌──────────────────────────────┐ │
│ │ c:¥vz                        │ │
  │ c:¥doc                       │
  │ c:¥tc                        │
  │ c:¥tc¥h                      │
  └──────────────────────────────┘
```

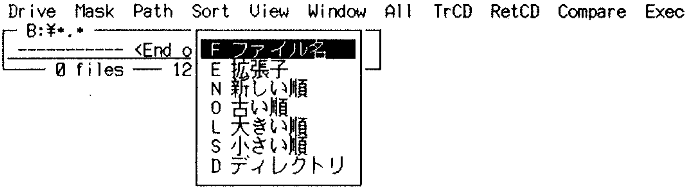
■ インスタントキー

Ver1.5から、ファイラーで次の記号キーを押していくつかの機能を簡単に実行できるようになりました。

[*]　　マスクを「*.*」に設定して再読み込みします。
[:]　　[D]と同様にドライブメニューを表示します。
[¥],[/] ルートディレクトリを表示します。

Sort　　　　　:ソート

ファイルを指定する順序でソートして表示します。[S]を入力すると、次のメニューを表示します。



ソートするモードを選択してください。選択したメニューには、「*」が付きます。もう一度「*」の付いたメニューを選択すると、ソートは解除します。

Ver1.5では、ソートを指定しない場合、ディレクトリの順番のままで表示します。従来のようにサブディレクトリを前にまとめて表示したい場合は、「D ディレクトリ」を選択してください。(この時、ディレクトリを読み込むのに余計に時間がかかります) ただし、マスクを設定した場合やソートを指定した場合は、サブディレクトリは前にまとめられます。

ソートの初期値は次のオプションで設定できます。

【*FSO*　:ディレクトリソートオプション 】

次の数値でソートモードを指定します。

1 = ファイル名
2 = 拡張子
4 = 新しい順
8 = 古い順
16 = 大きい順
32 = 小さい順
64 = ディレクトリ

View　　　　　:ビュー

[V]を入力すると、ディレクトリの表示モードをロングまたはショートに切り替えます。

＜ロングモード＞

```
Drive  Mask  Path  Sort  View  Window  All  TrCD  RetCD  Compare  Exec
 A:¥*.*
 VZ      .COM   45384  89-12-02  1:48
 VZ      .DEF   12255  89-12-02  1:48
 VZ      .DOC   64445  89-12-02  1:48
 VZS     .COM   37880  89-12-02  1:48
 VZS     .DEF   11928  89-12-02  1:48
 EZKEY   .COM    2770  89-12-02  1:48
 EZKEY   .DOC    9320  89-12-02  1:48
 VMAP    .COM    1940  89-12-02  1:48
 VMAP    .DOC    2326  89-12-02  1:48
 START   .BAT      76  89-12-02  1:48
 SAMPLE1 .TXT    2545  89-12-02  1:48
 SAMPLE2 .TXT     758  89-12-02  1:48
 README  .DOC   22821  89-12-02  1:48
 J31¥           <DIR>  89-12-02  1:48
 IBM¥           <DIR>  89-12-02  1:48
 SRC¥           <DIR>  89-12-02  1:48
 VZE     .DEF    9254  89-12-02  1:48
 UPDATE  .DEF     424  89-12-02  1:48
 HELP    .DEF    1709  89-12-02  1:48
 HELPE   .DEF    1587  89-12-02  1:48
 ----- 28 files -- 412672 bytes free ---

   C1   CU   CA   S1   SU    VOID  NWL  INS  REP  ^Z
```

＜ショートモード＞

```
Drive  Mask  Path  Sort  View  Window  All  TrCD  RetCD  Compare  Exec
 A:¥*.*
 VZ      .COM   VZ      .DEF   VZ      .DOC   VZS     .COM   VZS     .DEF
 EZKEY   .COM   EZKEY   .DOC   VMAP    .COM   VMAP    .DOC   START   .BAT
 SAMPLE1 .TXT   SAMPLE2 .TXT   README  .DOC   J31¥           IBM¥
 SRC¥           VZE     .DEF   UPDATE  .DEF   HELP    .DEF   HELPE   .DEF
 WS      .DEF   MI      .DEF   CMOS    .DEF   TOOL    .DEF   BLOCK   .DEF
 KEISEN  .DEF   ZENHAN  .DEF   GAME    .DEF   <End of Dir>
 -------------------------------- 28 files -- 412672 bytes free ---

   C1   CU   CA   S1   SU    VOID  NWL  INS  REP  ^Z
```

Window　　　　　:ウィンドウ

[W]を入力すると、もう1つのウィンドウを開き、カーソルを新しく開いたウィンドウへ移動します。もう一度実行すると、2つめのウィンドウを閉じます。(Ver1.5で仕様を変更しました)

```
Drive  Mask  Path  Sort  View  Window  All  TrCD  RetCD  Compare  Exec
 A:¥*.*                                      A:¥J31¥*.*
 VZ       .COM   45384  89-12-02  1:48      ..¥               <DIR>  89-12-02  1:48
 VZ       .DEF   12255  89-12-02  1:48      VZJ31    .DEF     12084  89-12-02  1:48
 VZ       .DOC   64445  89-12-02  1:48      FROLL    .COM      1375  89-12-02  1:48
 VZS      .COM   37880  89-12-02  1:48      VZSEL    .COM        65  89-12-02  1:48
 VZS      .DEF   11928  89-12-02  1:48      V        .BAT       338  89-12-02  1:48
 EZKEY    .COM    2770  89-12-02  1:48      VZJ31    .DOC      3458  89-12-02  1:48
 EZKEY    .DOC    9320  89-12-02  1:48      ---------- <End of Dir> ----------
 VMAP     .COM    1940  89-12-02  1:48      ---- 6 files -- 412672 bytes free ----
 VMAP     .DOC    2326  89-12-02  1:48
 START    .BAT      76  89-12-02  1:48
 SAMPLE1  .TXT    2545  89-12-02  1:48
 SAMPLE2  .TXT     758  89-12-02  1:48
 README   .DOC   22821  89-12-02  1:48
 J31¥            <DIR>  89-12-02  1:48
 IBM¥            <DIR>  89-12-02  1:48
 SRC¥            <DIR>  89-12-02  1:48
 VZE      .DEF    9254  89-12-02  1:48
 UPDATE   .DEF     424  89-12-02  1:48
 HELP     .DEF    1709  89-12-02  1:48
 HELPE    .DEF    1587  89-12-02  1:48
 ---- 28 files -- 412672 bytes free ----

  C1   CU   CA   S1   SU     VOID  NWL  INS  REP  ^Z
```

2ウィンドウ（ロング）

```
Drive  Mask  Path  Sort  View  Window  All  TrCD  RetCD  Compare  Exec
 A:¥*.*
 VZ       .COM   VZ       .DEF   VZ       .DOC   VZS      .COM   VZS      .DEF
 EZKEY    .COM   EZKEY    .DOC   VMAP     .COM   VMAP     .DOC   START    .BAT
 SAMPLE1  .TXT   SAMPLE2  .TXT   README   .DOC   J31¥            IBM¥
 SRC¥            VZE      .DEF   UPDATE   .DEF   HELP     .DEF   HELPE    .DEF
 WS       .DEF   MI       .DEF   CMOS     .DEF   TOOL     .DEF   BLOCK    .DEF
 KEISEN   .DEF   ZENHAN   .DEF   GAME     .DEF   <End of Dir>
                                  ---- 28 files -- 412672 bytes free ----

 A:¥J31¥*.*
 ..¥             VZJ31    .DEF   FROLL    .COM   VZSEL    .COM   V        .BAT
 VZJ31    .DOC   <End of Dir>
                                  ---- 6 files -- 412672 bytes free ----

  C1   CU   CA   S1   SU     VOID  NWL  INS  REP  ^Z
```

2ウィンドウ（ショート）

Ａｌｌ　　　　　　　　:オール

［A］を入力すると、ディレクトリの全ファイルに対して、選択してあるものは取り消し、選択してないものは選択します。全ての選択を取り消す場合は、［D］［↵］でディレクトリを再度読み込んでください。

ＴｒＣＤ　　　　　　　:Transfer CD

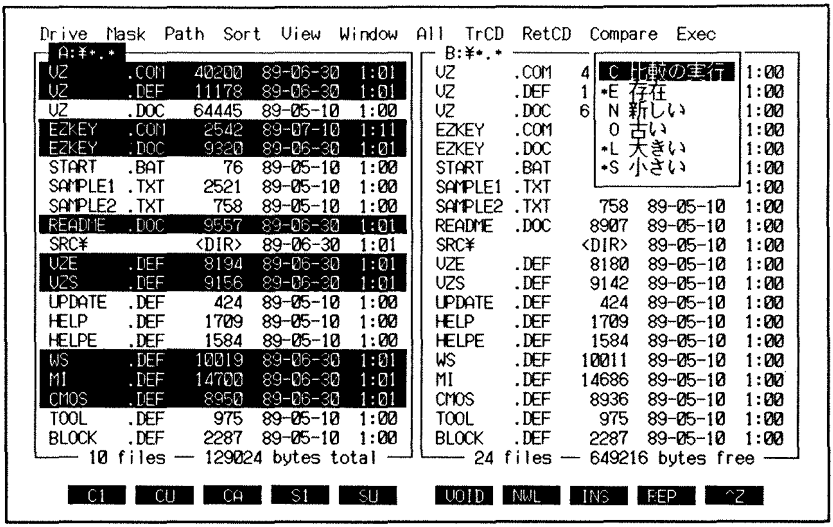
［T］を入力すると、現在表示しているディレクトリを、カレントドライブのカレントディレクトリとします。

ＲｅｔＣＤ　　　　　　:Retern to CD

［R］を入力すると、カレントドライブのカレントディレクトリを表示します。

Ｃｏｍｐａｒｅ　　　　:コンペア

2つのウィンドウを表示している際に、ソースディレクトリ（カーソルがある方のディレクトリ）の内容を、指定オプションでもう一方のディレクトリ（ターゲットディレクトリ）の内容と比較し、結果を選択ファイルとします。［C］を入力すると、次のメニューを表示します。



2番目以降のメニューで、比較するオプションをON／OFFします。標準設定では、「E 存在」と「N 新しい」の2つのオプションが選択されており、この場合は、

- ソースディレクトリのみに存在するファイル
- ソースディレクトリの方が日付が新しいファイル

を選択します。「C 比較の実行」メニューを選択すると、比較を実行します。

Exec　　　　　:実行

■ 選択したファイルがない場合

[E]を入力すると、ただちにコマンドラインモードへ移行します。コマンドラインモードは、DOSのコマンドを実行するモードです。詳しくは次章をご覧ください。

■ 1つ以上のファイルを選択している場合

選択したファイルに対して、順次DOSコマンドを実行します。[E]を入力すると、次のメニューを表示します。

```
Drive  Mask  Path  Sort  View  Window  All  TrCD  RetCD  Compare  Exec
┌ B:¥*.* ─────────────────────────────────────┐
│ VZ       .COM   45384  89-12-02   1:48 │989 by c.mos          ┌──────────────┐
│ VZ       .DEF   12255  89-12-02   1:48 │                      │ C コピー      │
│ README   .DOC   22821  89-12-02   1:48 │                      │ D 削除        │
│ ----------- <End of Dir> ----------- │                      │ P パック      │
└─── 3 files ── 81920 bytes total ───┘                      │ U アンパック  │
                                                              │ V パックリスト│
                                                              │ A 比較(Asc)   │
                                                              │ B 比較(Bin)   │
                                                              │ T テスト      │
                                                              │ X 実行        │
                                                              └──────────────┘
```

ここで、実行したいコマンドを選択してください。すると、ファイラーを終了し、コマンドラインモードへ切り替わり、実行するコマンドを表示します。複数のファイルを選択している場合は、ファイル名が置き換わるところが「%?」になります。ここで、[↵]を入力すると、選択したファイルに対して、指定のDOSコマンドを順次実行します。

コマンドラインモードから抜ける場合は、

・ファイラーへ ········ [SHIFT]+[↵]

・エディタへ ·········· [ESC]

を入力してください。

★注意★

Execメニューの中の次のコマンドは、それぞれ実行するコマンドが必要になります。

lharc. exe

P パック

U アンパック

V パックリスト

fc. exe

A 比較(Asc)

B 比較(Bin)

5.4 複数ファイルのコピー

ファイラーの「Execメニュー」を利用すると、選択したファイルに対してコピーや削除などあらかじめ設定したコマンドを順次実行することができます。では、実際にファイルをコピーする例で説明しましょう。

[例]

・ファイラーを起動します。(A>vz[↵][↵]等)

・いくつかコピーしたいファイルを選択します。

・[**E**]を押し、Execメニューを表示します。

・[**C**]でコピーを選択します。

すると、コマンドラインモードへ移り、コマンドラインにコピーコマンドを表示します。(ファイル名のところは「%?」になっています)

・そのまま、コピー先のディレクトリを入力します。

・[↵]を押すと、順番にコピーされていきます。

・コピーが終了したら、[**SHIFT**]+[↵]でファイラーに戻ります。

このようにして、複数ファイルのコピーを行なうことができます。

```
A:¥>copy VZ.COM b:
        1 個のファイルをコピーしました.

A:¥>copy VZ.DEF b:
        1 個のファイルをコピーしました.

A:¥>copy README.DOC b:
        1 個のファイルをコピーしました.

A:¥>
    C1  CU  CA  S1  SU    VOID  NWL  INS  REP  ^Z
```

Execメニューに設定されている各コマンドの機能について説明します。

■C コピー

「copy」コマンドを呼び出し、ソースディレクトリで選択したファイルを、ターゲットディレクトリへコピーします。ただし、1つのディレクトリしか表示していない場合は、ソースディレクトリからカレントディレクトリへコピーします。わざわざターゲットディレクトリを表示しなくとも、いったん選択して、コマンドラインでコピー先のディレクトリを入力してもよいでしょう。

■D 削除

「del」コマンドを呼び出し、選択したファイルを消去します。

■P パック

ソースディレクトリで選択したファイルを、ターゲットディレクトリの指定ファイルへ圧縮して格納します。ターゲットディレクトリでファイルを指定（1つのファイルを選択）していない場合、およびソースディレクトリしか表示していない場合は、メニュー選択後のコマンドラインモードにて、カーソルを「a」の後ろまで戻し、圧縮先のファイル名を指定しなければなりません。

※　圧縮に指定しているlharcは、通信ネットなどで入手可能なPDSです。

■U アンパック

ソースディレクトリで選択した圧縮ファイルを解凍します。

■V パックリスト

ソースディレクトリで選択した圧縮ファイルの内容を表示します。Ver1.5では、拡張子による自動実行機能が追加され、「.lzh」の拡張子をもつファイルを[↵]で選択すると、ただちにこのコマンドを実行することができます。詳しくは『6.3 ファイラーとの連携』で説明します。

■A 比較(Asc)

「fc」コマンドを呼び出し、ソースディレクトリで選択したファイルを、ターゲットディレクトリの同じ名前のファイルとアスキーモードで比較します。

■B 比較(Bin)

「fc」コマンドを呼び出し、ソースディレクトリで選択したファイルを、ターゲットディレクトリの同じ名前のファイルとバイナリーモードで比較します。

■T テスト

「dir」コマンドを呼び出し、ソースディレクトリで選択したファイルを表示します。このメニューは、テスト用に用意されています。

■X 実行

選択したファイル名を、そのままコマンドラインへ渡します。Ver1.5では、拡張子による自動実行機能が追加され、「.com .exe」の拡張子をもつファイルを[↵]で選択しても、コマンドラインへ渡して実行することができます。

以上のコマンドは、DEFファイルの次の箇所で定義してありますので、自由に設定できます。

```
* S システムメニュー

8       "",15,9
        "C コピー      copy %1 %2"
        "D 削除        del "
        ...
```

このメニューの書式は、次のとおりです。

[書式]

```
8       "",<width>,<items>
        { "<title><ﾀﾌﾞ><command>" }
```

<width>	：メニューの横サイズ
<items>	：メニューの個数
<title>	：各コマンドのタイトル
<ﾀﾌﾞ>	：1つ以上のTABコード
<command>	：コマンドの記述

<command>中の次の文字が、ファイラーで選択したファイル名、またはパス名に置き換えられます。

%1	ソースディレクトリで選択した各ファイル名 （カレントディレクトリ以外の場合は、パス名に続き）
%2	ターゲットパス名（※） （ファイルが選択されている場合には、そのファイル名）
%*	ソースディレクトリで選択した全てのファイル名の並び

※ 2ディレクトリ表示の場合の、カーソルの無い方のディレクトリです。

Ver1.5では、終わりの「%1」は省略できます。
[例]

del %1" → del "

<command>記述中には、通常のコマンドラインモードからの入力と同様、エイリアス、マルチステートメント等を使用できます。

★注意★
「%*」で、選択したファイルを並べて入力する場合、あまり長すぎると後ろの部分がカットされます。また、この場合は必ずカレントディレクトリで行なってください。さもないと、各々のファイルにパス名が付加され、いっそう長くなってしまいます。

第6章 DOSコマンドの実行

6.1 コマンドラインモード

6.2 エイリアス

6.3 ファイラーとの連携

6.4 コンソール出力の取り込み

6.1 コマンドラインモード

編集中にDOSのコマンドを実行したい場合はよくあります。VZでももちろんできます。次の章では、これをもう一歩進めて、VZの中でDOSのコマンドを実行するのではなく、DOSのコマンドラインとVZを自由に往来する方法について説明します。これを「常駐モード」と呼びます。この章では、まずDOSコマンドを実行する際の種々の拡張機能について説明しておきます。これらの機能は、常駐しているかどうかによらず使用することができます。

[ESC] E　　　　　　　　:DOSコマンド実行

編集時にこのコマンドを実行すると、DOSの画面を表示し、最下行に黄色で、

《A:¥>》

というようにプロンプトを表示します。この状態を、「コマンドラインモード」と呼びます。コマンドラインモードでは、次の種々の機能を利用して、DOSコマンドを実行することができます。

- 行編集・ヒストリー機能
- エイリアス機能
- マルチステートメント機能
- 環境変数の参照
- 選択ファイルに対するコマンドの連続実行

また、前章で述べたように、ファイラーから「Execメニュー」を選択することによっても、コマンドラインモードへ移ることができます。

コマンドラインモードの操作コマンド

コマンドラインモードでは、他の文字列入力時と同様に、行編集・ヒストリー機能が使用できます。その他にも次のコマンドを実行できます。

[ESC] :エスケープ

編集モードへ戻ります。この時のDOS画面は保存され、次回コマンドラインモードに移る際に、再び表示します。

---

^M [↵] :リターン

入力したDOSコマンドを実行します。実行が終了すると、ただちにコマンドラインモードに戻ります。

[↵]のみ入力した場合は、「command.com」を呼び出します。画面には

```
EXITでエディタへ戻ります
Command ﾊﾞｰｼﾞｮﾝ x.xx
A>
```

というメッセージが出て「command.com」に制御が渡ったことを示します。この状態ではVZは働いていませんので、コマンドラインでVZの種々の機能を使うことはできません。再びVZに戻るときは、

```
A>exit[↵]
```

を実行してください。「command.com」を指定パスからロードできない場合、または実行するメモリが不足している場合は、

```
《COMMAND.COMを起動できません》
```

と表示し、ただちに編集モードに戻ります。

★注意★

コマンドラインで、もう一度VZを起動してはなりません。

[SHIFT]+[↵]　　　　:ファイラー

ファイラーを起動します。[SHIFT]+[↵]の前に文字列を入力すると、いろいろな動作をしますが、これについては『6.3ファイラーとの連携』で説明します。

[SHIFT]+[ESC]　　　　:DOS画面出力の参照

コンソール画面を表示します。これについては『6.4コンソール出力の取り込み』で説明します。

^W　　　　:エイリアス一覧

エイリアスの定義の一覧を表示します。エイリアスについては『6.2エイリアス』で説明します。

[CTRL]+[↵]　　　　:コマンドの展開再表示

入力したDOSコマンドを、エイリアス・環境変数で展開してコマンドラインに再表示します。実行はしません。このコマンドで、エイリアスや環境変数がどのように展開されるかを事前に知ることができます。

コマンドライン入力の拡張機能

コマンドラインでDOSのコマンドを入力する場合には、次の拡張機能が利用できます。

■ エイリアス機能

DOSコマンドを、短縮した別の名前で入力する機能です。次節で詳しく述べます。

[例]

```
A:¥>ls[↵]
→  dir /w
```

■ マルチステートメント

複数のコマンドを、「<ｽﾍﾟｰｽ>;」で区切って、一行に記述できます。

[例]

```
A:¥>del ¥tmp ;rd ¥tmp[↵]
→  del ¥tmp
   rd  ¥tmp
```

指定したコマンドを順次実行します。コマンドの実行を中断する時は、[**STOP**]キー（J-3100版、IBM PC版では[**Ctrl**]+[**Break**]キー）を押してください。^C だとその時実行しているコマンドは中断しますが、次のコマンドを実行してしまいます。

■ 環境変数の参照

バッチファイルと同様に、「%...%」で環境変数を参照できます。

[例]

```
A:¥>optasm %opt%[↵]
→ optasm /ml
```

以上をまとめると、入力したコマンドは、次の順序で展開し、実行します。

```
[エイリアス展開] → [環境変数展開] → [再表示] → [各コマンドの実行]
                                              ↑
                                     [CTRL]+[↵]の場合は、ここで中断
```

※「オプション***Sd***」を「-」に指定すると、再表示しないで実行することができます。

【 ***Sd+*** :DOSコマンドラインの展開再表示 】

6.2 エイリアス

エイリアス（別名）機能は、コマンドを別の短い名前で定義しておき、より少ないキーストロークで実行する機能です。エイリアスは、DEFファイルの

```
* A エイリアス
```

の箇所で定義します。どのようなエイリアスが定義されているか知りたい場合は、^W を実行してください。

```
A>vz
VZ Editor Version 1.50  Copyright (C) 1989 by c.mos






┌ 【エイリアス一覧】 ─────────────────┐
│ cat      type                        │
│ ls       dir %-w %1                  │
│ sd       symdeb                      │
│ rene     ren %1 %<.%2                │
│ fca      fc %1 %2%¥%>                │
│ fcb      fc /b %1 %2%¥%>             │
│ sete     set %1=%%1%                 │
│ test1    echo %: ; echo %& ; echo %. │
│ test2    echo %wd% %%a %1 %%1% %%a%  │
└──────────────────────────────────────┘
A:¥>
    C1  CU  CA  S1  SU    VOID  NWL  INS  REP  ^Z
```

DEFファイルにおけるエイリアス定義の書式は、次のとおりです。

[書式]

〈エイリアス名〉〈コマンド〉

[例]

```
cat     type
ls      dir %-w %1
```

最初の例のように単にコマンド名を置き換えるだけでなく、2つ目の例のように、引数を参照したり、マルチステートメントを利用して複数のコマンドを記述することもでき

ます。ただし、必ず1行内で記述しなくてはなりません。

バッチファイルでは、バッチファイルを起動したパラメータを「%1～%9」で参照することができますが、VZのエイリアス機能では、もう少し複雑な参照が可能です。

%1～%9	1～9番目のパラメータ
%*	全パラメータ
%*n	n 番目以降の全パラメータ
%:	パス名
%&	ファイル名
%.	拡張子
%<	パス名＋ファイル名
%>	ファイル名＋拡張子
%:n	n 番目のパラメータのパス名
%¥	直前の文字が¥で無い場合にのみ¥を付加
%-	スイッチキャラクタ (/,-)
%%	% 文字

例えば引数が「a:¥path¥file.ext」の場合、

```
%:      :a:¥path¥
%&      :file
%.      :ext
%<      :a:¥path¥file
%>      :file.ext
```

というように、パスの一部を参照することができます。

[例]

(定義) rene ren %1 %<.%2

A:¥>rene test.asm bak[CTRL]+[↵]

→ ren test.asm test.bak

エイリアスは1度しか展開されません。従ってコマンドの記述の中に、他で定義されたエイリアスを記述しても、それは展開されません。このため、次の例のような重複定義が可能となっています。

[例]

(定義) dir dir %-w %1

実行中にエイリアスを再定義する場合は、DEFファイルをオープンし、エイリアス定義の箇所を編集したうえで、

^K^	:再カスタマイズ

コマンドを実行してください。このコマンドは、『第10章 マクロ機能』で詳しく説明します。ただしエイリアス定義を追加する場合は、あらかじめ次のオプションで、余分なバッファを確保しておかなければなりません。

【 ***Ba*** 0 :エイリアスバッファ 】

6.3 ファイラーとの連携

コマンドラインから[SHIFT]+[↵]でファイラーを起動することができるのは、前にも触れました。この時、次のように[SHIFT]+[↵]の前に、表示したいディレクトリのパスおよびマスクを入力することで、ただちにそのディレクトリを開くことができます。
[例]

A:¥>b:[SHIFT]+[↵]
A:¥>c:¥doc[SHIFT]+[↵]
A:¥>.def[SHIFT]+[↵]

この時は、文字列の中にスペースを入れてはなりません。

さて、前章の『5.4複数ファイルのコピー』のところで、ファイラーのExecメニューにより、コマンドを連続して実行する方法について説明しました。しかし、この方法だと、あらかじめ設定したコマンドしか実行することができません。(コマンドラインで再編集すればできます)
Ver1.5では、この部分を改良しました。コマンドラインでコマンドを打ちこみ、ファイラーを起動して[↵]でファイルを選択することで、選択したファイル名をコマンドラインに入力することができます。
[例]

A:¥>copy [SHIFT]+[↵] ("copy"の後にスペースを入れます)
・ファイラーで、例えば「readme.doc」へカーソルを移動して[↵]
A:¥>copy readme.doc

このように、入力文字列の中にスペースが含まれる場合は、コマンドとみなすわけです。
さらに、次のような記述もできます。
[例]

A:¥>copy b:¥[SHIFT]+[↵]
・「b:¥」のディレクトリを表示します。
・複数ファイル、例えば「vz.com vz.def」を選択して[↵]
A:¥>copy b:¥%?

ここで[↵]を押すと、複数ファイルのコピーが実行されます。行末からスペースまでをパス・マスクとみなすわけです。

拡張子による指定コマンド実行

ファイラーで「vz.com」などのバイナリーファイルを選択すると、従来は、そのままオープンしてしまい、画面にゴミが表示されました。Ver1.5では、これを避けるとともにもう一歩進んで、特定の拡張子を選択すると、指定したコマンドを実行すべくコマンドラインモードへ移る機能を設けました。

[例]

・ファイラーで、「vz.com」へカーソルを移動して[↵]

```
A:¥>vz.com
```

・同じく、「pds.lzh」へカーソルを移動して[↵]

```
A:¥lharc v pds.lzh
```

この機能は、DEFファイルに次の箇所にメニューとして記述します。

```
* S システムメニュー

9       "",0,3
        "com"
        "exe"
        "lzh lharc v "
```

メニューの書式は、次のとおりです。

[書式]

```
9       "",0,<items>
        { "<ext>[<ｽﾍﾟｰｽ><command>]" }

<items>         : メニューの個数
<ext>           : 拡張子
<command>       : コマンドの記述
```

<command>は、Execメニューと同様です。

★注意★

この機能は、複数のファイルを選択している場合には働きません。

6.4 コンソール出力の取り込み

VZでは、編集するテキスト以外に、「コンソールファイル」をあらかじめオープンしており、DOSコマンド実行中の画面出力を、自動的にそこへ取り込むことができます。そして、編集モードおよびコマンドラインモードで次のコマンドを実行すると、「コンソールファイル」を参照できます。

[**SHIFT**]+[**ESC**]　　　: DOS画面出力の参照

```
 P □      1:1                                  0R console
dir↓
↓
 ドライブ A: のディスクのボリュームラベルはありません.↓
 ディレクトリは A:¥↓
↓
VZ       COM     45384  89-12-02   1:48↓
VZ       DEF     12255  89-12-02   1:48↓
VZ       DOC     64445  89-12-02   1:48↓
VZS      COM     37880  89-12-02   1:48↓
VZS      DEF     11928  89-12-02   1:48↓
EZKEY    COM      2770  89-12-02   1:48↓
EZKEY    DOC      9320  89-12-02   1:48↓
VMAP     COM      1940  89-12-02   1:48↓
VMAP     DOC      2326  89-12-02   1:48↓
START    BAT        76  89-12-02   1:48↓
SAMPLE1  TXT      2545  89-12-02   1:48↓
SAMPLE2  TXT       758  89-12-02   1:48↓
README   DOC     22821  89-12-02   1:48↓
J31          <DIR>      89-12-02   1:48↓
IBM          <DIR>      89-12-02   1:48↓
SRC          <DIR>      89-12-02   1:48↓
VZE      DEF      9254  89-12-02   1:48↓
UPDATE   DEF       424  89-12-02   1:48↓
   ファイル 窓換 文換 窓割 記憶   検索 置換 カット インサート ブロック
```

コンソールファイルは、画面最上部のステータスラインが黄色（PC-9801版）で、ファイル名は「console」となります。再度[**SHIFT**]+[**ESC**]を実行すると、コンソールファイルはクローズしますが、その内容は保存されます。

エディタを起動した直後は、コンソールファイルには何も入っていません。[**ESC**] **E**コマンドを実行して、コマンドラインモードからDOSコマンドを実行すると、その画面出力がコンソールファイルに書き出されていきます。

■ コンソールバッファ

コンソールファイルは、他のテキストと全く同様に扱うことができます。ただし、バッファのサイズは最大時32KBで、それ以上は大きくはなりません。一杯になった場合には、先頭から1/4づつ切り詰められます。サイズは「オプション***Bo***」で設定します。

【 ***Bo***32 :コンソール出力取込みバッファサイズ (0..32KB) 】

このオプションを「0」にすると、コンソールファイル機能は使用できなくなります。

コンソールバッファは、グラフィックVRAMやEMS上に確保されますが、これらを搭載していない機種ではメインメモリ上に取られますので、メモリサイズが大きくて困る場合には、バッファのサイズを小さく設定してください。

■ コンソールファイルの使い方

コンソールファイルは、次のような時に利用できます。

- dir等のコマンドの出力をテキスト中に貼り込みたい
- コンパイルエラーからタグジャンプしたい

これらの機能は、従来いったんファイルにリダイレクトしてから、エディタで読み込み、参照していましたが、VZのコンソールファイル機能を利用すれば、そんな必要はありません。

■ コンソールファイルの取り込みの停止

次のオプションを「-」に指定すると、一時的に画面出力の取り込みを停止します。

【 ***Sx***+ :コンソール出力の取込み 】

■ 特殊なコードの取り込み

画面出力のうち「エスケープシーケンスコード（1Bhに続く種々の画面制御を行なうコード列」は、コンソールファイルには取り込まれません。これは、DOSのプロンプトを

```
A>prompt $e[22m$p$e[m$g
```

というように設定している場合、プロンプトを表示するたびにエスケープシーケンスコードをコンソールファイルに出力し、たいへん見苦しいためです。しかし、「オプション***Se***」を「-」に指定することで、エスケープシーケンスコードも取り込みよう指定できます。

【 ***Se***+ :エスケープシーケンスのスキップ 】

また、コマンドによっては、1行内で次々と表示を書き換えていくものがあります。(例:format) これらの出力をそのまま取り込むと、コンソールファイルに改行の無い長い1行が作られてしまい、編集に支障をきたす場合があります。これを避けるために、CRコード(0Dh)のみが送られてきた場合は、1行分の内容をクリアします。

さらに、BSコード(08h)が送られてきた場合、直前の文字を1文字削除します。

■ バッファリング

Ver1.5で、コンソールバッファをEMS上に確保することができるようになりました。このため、1文字取り込むたびにEMSメモリをマッピングしたのでは、文字の出力がたいへん遅くなります。これを避けるため、メインメモリ上に約100バイトのバッファを設け、LFコード(0Ah)が送られてくるか、あるいはバッファが一杯になった時点でコンソールバッファの方へ出力するようにしています。この一時記憶バッファは、VZのコマンドパラメータエリア(0080h~00FFh)に確保しますので、起動時のパラメータを余り長くするとバッファの容量が小さくなり、動作に支障をきたします。

■ 編集ファイル指定でのオープン

Ver1.5から、編集ファイル指定で「console」を入力することで、コンソールファイルをオープンできるようになりました。

第7章　より高度な使い方

7.1 常駐モード

7.2 EMS

7.3 スワップ

7.1 常駐モード

エディタのコマンドラインモードからDOSコマンドを実行すると、コマンドを実行するたびに「command.com」を呼び出すために、一定のタイムラグが発生します。また、通常DOSのコマンドラインでは、特別のツールを常駐させない限り、ヒストリー等の編集機能は使用できません。ましてや、VZのファイラーのようにファイルを選択してコピーしたい場合には、専用のアプリケーションを起動しなければなりません。

これらの問題点を解消し、より緊密なコマンドラインとの往来を可能とする機能が「常駐モード」です。VZが常駐すると、ちょうどVZのコマンドラインモードがDOSのコマンドラインへ拡張されたかのように、VZの種々の機能をコマンドラインで実行することができます。

常駐モードの起動と終了

VZを「常駐モード」で使用したい場合は、起動時パラメータで「オプション**Z**」を指定してください。

[例]

```
A>vz -z
VZ Editor Version 1.5x  Copyright (C) 1989 by c.mos
メモリに常駐しました. (-zで解放)

A>
```

これでVZはメモリに常駐しました。付属の「vmap.com」を実行すると、VZが常駐している状態を知ることができます。

[例]

```
A>vmap
VMAP Version 1.0x  Copyright (C) 1989 by c.mos

addr PSP  blks   size  owner/parameters          hooked vectors
---- ---- ---- ------  ------------------------- -------------
...
xxxx xxxx    1   xxxxx  vz -z                     06 21 29
...
```

常駐をやめ、メモリを解放したい場合は、再度「vz -z」を実行してください。
[例]

```
A>vz -z
VZ Editor Version 1.5x  Copyright (C) 1989 by c.mos
メモリを解放しました.

A>
```

★注意★
テキスト編集中に、メモリを解放しようとしても、

```
編集中のテキストがあります.
```

と表示され、メモリを解放できません。

常駐モードでのVZの起動

VZを常駐した後のコマンドラインは、VZが受け持ちます。したがって、非常駐時に[ESC]Eで移行するコマンドラインモードと同様に、行編集・ヒストリー機能、およびエイリアス・マルチステートメント等の拡張機能が使用できます。

エディタを起動する時は、[ESC]を押してください。次のように入力ファイルウィンドウが開きます。

```
J31              <DIR>      89-12-02    1:48
IBM              <DIR>      89-12-02    1:48
SRC              <DIR>      89-12-02    1:48
VZE        DEF       9254   89-12-02    1:48
UPDATE     DEF        424   89-12-02    1:48
HELP       DEF       1709   89-12-02    1:48
HELPE      DEF       1587   89-12-02    1:48
WS         DEF      10019   89-12-02    1:48
MI         DEF      14682   89-12-02    1:48
CMOS       DEF      10535   89-12-02    1:48
TOOL       DEF        975   89-12-02    1:48
BLOCK      DEF       2287   89-12-02    1:48
KEISEN     DEF       3762   89-12-02    1:48
ZENHAN     DEF       1023   89-12-02    1:48
GAME       DEF       2573   89-12-02    1:48
       28 個のファイルがあります.
   412672 バイトが使用可能です.

A>vz -em -z
VZ Editor Version 1.50  Copyright (C) 1989 by c.mos
┌ 【入力ファイル】 <A:¥> ──────────────┐
│                                        │
└────────────────────┘
A>
     C1   CU   CA   S1   SU      VOID  NWL  INS  REP  ^Z
```

また、非常駐時と同様に、

A>vz [<編集ファイル指定>]

でエディタを起動することもできます。(オプションやDEFファイルの指定はできません。) いずれの場合も、VZ本体のロード、およびカスタマイズはいたしませんので、瞬時にエディタが起動します。

コマンドラインの実行キーをもう一度まとめておきます。

[ESC]	:エディタの起動
[SHIFT]+[ESC]	:DOS画面出力の参照
^M [↵]	:コマンド実行
[CTRL]+[↵]	:コマンドの展開再表示
[SHIFT]+[↵]	:ファイラー
^W	:エイリアス一覧

★注意★

常駐時、ファンクションキーに設定されているテンプレート機能は使用できません。代わりに、VZの編集機能をお使いください。

常駐時のメモリ消費

「vmap」を実行すると、VZが常駐した際のメモリ消費量を知ることができます。これは、標準設定でEMSを搭載していない機種では、110KBを越える大きなものです。これでは、大きなアプリケーションがメモリ不足で起動できなくなってしまいます。EMSが利用できる場合は、後ほど『7.3スワップ』で述べますが、とりあえず「オプションSW3」を指定して常駐してみてください。

```
A>vz -sw3 -z
VZ Editor Version 1.5x  Copyright (C) 1989 by c.mos
EMSを xx ページ使用します.
メモリに常駐しました. (-zで解放)
```

再び「vmap」を実行すると、常駐メモリが2.5KB弱に減っていることと思います。これは、VZの占有メモリをEMSに一時待避（スワップアウト）しているためです。

ここでは、EMSを使えない場合の常駐メモリの減らし方について説明します。

■　裏グラフィックVRAMをワークとする

（PC-9801ノーマルモードのみ）

裏グラフィクVRAMを使用するアプリケーションを使わない場合は、次のように「オプション**GV**」に「1～3」の値を設定して、裏グラフィクVRAMをワークに指定します。

```
A>vz -gv1 -z
```

こうすると、常駐メモリが約53KBになります。裏グラフィックVRAMを他のソフトが使用する場合でも、それが A8000h ～ B8000h の64KBである場合は、「オプション**GV**3」を指定することで、その上位の64KB（B8000h ～ C0000h、E0000h ～ E8000h）をワークに使用できます。ただしこれは、16色表示の可能な機種に限られます。

■　コンソールバッファを縮小する

裏グラフィックVRAMが使用できない状況にある場合は、次のように、「オプション**Bo**」を小さくすると、その分常駐サイズが減ります。

```
A>vz -bo0 -z
```

また、編集テキストをオープンしている状態では、その分のメモリが消費され、常駐サイズが拡大します。

7.2 EMS

昨今、アプリケーションはどんどん肥大化し、640KB程度のメインメモリでは限界に達してきました。この打開策として考案されたのが、EMS（Extended Memory System）です。EMSは、メインメモリの外に、通常64KBのウィンドウを設け、これに16KB／ページのメモリをマッピングして使用します。

VZも、Ver1.5からEMS対応となりました。EMSを使うことによって最も期待されることは、常駐メモリが小さくなることです。しかし、VZのような中規模のアプリケーションにとって、これは意外に大変なことです。というのは、仮にVZの本体をEMS上におけば、常駐サイズは非常に小さくなりますが、その場合テキストバッファ、GVRAM用ワーク等でEMSを使用できなくなってしまいます。これではせっかくのEMSメモリを有効に使うことができません。結局VZでは、次節で述べる「スワップ」の手法を巧妙に使って、見かけ上常駐サイズを小さくしています。

■ EMSメモリの割り当て

標準設定では、EMSメモリが存在する場合、最大1MBまでVZが使用します。しかし、複数のアプリケーションでEMSメモリを分けて使用する場合は、次のように「オプション***EM***」でVZで使用するEMSのページ数を設定してください。

【 ***EM***　：EMSの使用ページ数 】
EM0　：EMSメモリは使用しない。
EMn　：EMSメモリを n ページ使用する。(n=0～64)
EM　：EMSメモリの残りを全て使用する。(最大64ページ)

[例]

```
A>vz -em8 -z
VZ Editor Version 1.5x  Copyright (C) 1989 by c.mos
EMSを　8 ページ使用します.
メモリに常駐しました. (-zで解放)
```

■ EMSのページフレーム

VZでは、64KBの連続したページフレームを必要とします。また64KBより大きいページフレームには対応していません。

PC-9801RA,RXでは、GVRAMと重なったアドレスにページフレームを割り当てる機能をもっています。VZはこの機能に対応しています。

■ EMSメモリの使用目的

EMSは、以下のワークとして使用します。

(A) 画面セーブエリア、ディレクトリーバッファ等、従来「オプション **GV**」で裏グラフィックVRAMを使用していたワーク
…… 2ページ
(B) コンソールバッファ …… 0～2ページ
(C) 編集テキストバッファ………… テキスト毎に1～2ページ
(D) テキストスタック……………… 2ページ
(E) スワップ領域…………………… コード用4ページ、テキスト用に残り全て

(A), (B)は、**GV**1～3を指定した場合は、従来どおりGVRAMを使用します。また、(C), (D)は、メインメモリがある場合には使用しません。メインメモリが一杯になると、自動的にEMSを使用します。「オプション **Bf**」で次のようにフリーエリアのサイズを指定することにより、強制的にEMS上に編集バッファを確保することも可能です。

【 **Bf**32 :DOSコマンド用メモリの最小値 】
Bf0 : メモリの上限までエディタで使用する。
Bfn : フリーエリアを n KB確保する。(デフォルト32KB)
Bf : 編集バッファはEMS上にのみ確保する。

※ テンポラリファイルをEMS上に作成することはできません。

■ EMSドライバのバージョン

EMSドライバには、現在バージョン3.2とバージョン4.0があります。VZは、バージョン3.2のドライバでも動作します。ただし、バージョン4.0のもつ機能である「ハンドル名の設定」は使用しており、VZが使用するEMSのハンドルとして、VZのコマンド名を与えています。

[例]

```
A>vmap
VMAP Version 1.0x  Copyright (C) 1989 by c.mos
...

handle page   size  name     (EMS frame:C000h)
------ ---- ------  --------
     1  100  1600k  IOS10
     2   27   432k  VZ
  free    0     0k
 total  127  2032k
```

■ その他の機能

次のメニューを表示すると、メニューの一番下の行に、VZが管理しているEMSのページ数とあきページ数を表示します。

[SHIFT]+[f・2]（ｻﾌﾞ）　　:サブメニュー

7.3 スワップ

「スワップ（swap）」は「交換」を意味します。アプリケーションで「スワップ機能」といった場合は、アプリケーション動作中にDOSコマンドを実行する際、使用中のメモリをいったんEMSやディスクにセーブしてその分フリーメモリ拡大し（スワップアウト）、コマンド実行終了後にその部分をロード（スワップイン）して実行前の状態に戻す機能を指します。

VZのVer1.5もスワップ機能をもちます。テキスト編集中にDOSコマンドを実行する際、スワップを指定すると、テキストバッファの内容をEMSまたはディスクに出力し、フリーエリアを拡大します。これによって、たくさんのプログラムを同時にエディットしつつ、コンパイラ等大きなフリーエリアを必要とするコマンドを実行することができるようになりました。

さらに、VZではコード領域までもスワップアウトすることができます。この場合は、「ローダー部」が2.5KB弱メモリに残るだけになります。この機能を常駐後ただちに実行することで、常駐サイズをこの値にまで小さくできるわけです。こうすると、コマンドラインへ移るたびにVZのコード領域を再ロードしますから、その分タイムラグが生じますが、システムが十分高速な場合にはほとんどわかりません。これは、巧妙に必要な部分だけロード／セーブしているためです。

■　スワップモード

スワップにはいくつかのモードが設定できます。これは「オプション***SW***」を用います。

【 ***SW***1 :スワップモード 】
SW0　：スワップ機能を使用しない。
SW1　：テキスト領域をスワップアウトする。（デフォルト）
SW2　：コード領域も含めてスワップアウトする。
SW3　：常駐時にあらかじめコード領域までスワップアウトする。

［例］

```
A>vz -sw3 -z
```

システムの性能と、必要メモリサイズに応じて、上記のオプションを設定してください。ただし、「オプション***SW***」は実行中に変更してもかまいません。この場合は、[**ESC**]@コマンドで変更します。

★注意★

EMSもGVRAMもワークとして使用できない場合は、コード領域のスワップアウトはできません。

■ EMSとディスクの消費

スワップは、「コード領域」→「テキストバッファ」の順で行なわれます。EMSメモリがある場合は、それが無くなるまで使用し、残りはテンポラリファイルとして出力します。

■ 強制スワップ指定

「**SW**3」でテキスト編集時、または「**SW**1,2」指定時は、そのままではスワップは実行しません。これは、たくさんのテキストを読み込んでいて、「dir」や「ren」など、メモリを確保する必要のないコマンドを実行するたびにスワップアウトしていたのでは、無駄な時間を生じるからです。このため、テキスト編集時には次の方法でスワップアウトを指定してください。

「コマンドの先頭文字を大文字で指定する」

[例]

```
A>vz -sw3 -z
(テキストをオープンし、[ESC]Eでコマンドラインへ)

A>vmap
VMAP Version 1.0x  Copyright (C) 1989 by c.mos

addr PSP  blks   size  owner/parameters          hooked vectors
---- ---- ---- ------  ------------------------  -------------
...
2618 2618    2  [69968]  vz -sw3 -z                 06 21 29
372F         1  429328   <free>

F:¥VZ>Vmap
VMAP Version 1.0x  Copyright (C) 1989 by c.mos

addr PSP  blks   size  owner/parameters          hooked vectors
---- ---- ---- ------  ------------------------  -------------
...
2618 2618    1  [2448]  vz -sw3 -z                 06 21 29
26B2         1 496864  <free>
```

※　各サイズはこの値とは限りません。

従って、スワップアウトを必要とするコマンドは、次のようにエイリアスに登録して

おくとよいでしょう。
[例]

　　　* A エイリアス

　　　tc Tc

　また、非常駐時に、[ESC] Eのコマンドラインで[↵]のみ入力して「command.com」を呼び出した場合も、スワップを実行します。

■　常駐時の制限事項

　常駐時には、

　　　「編集テキストの占めるメモリは、連続した領域でなければならない」

という制限があります。これに対して、常駐した際に取得したVZ本体＋ワークのメモリと、編集テキストの占めるメモリとは、分割されても構いません。このために、常駐後、さらに他のソフトを常駐する場合は、必ず編集テキストを読み込んでいない状態で行なってください。

　また、「**SW**3」で常駐した場合は、「symdeb」や「edlin」等DOSの1行入力ファンクションを利用するアプリケーションの上からは、VZを起動できなくなります。これは、VZのコード領域をロードするメモリが確保できないためです。このような時にもVZを起動したい場合には、あらかじめ

　　　[SHIFT]+[ESC]　[ESC] E

を実行してください。こうすると、コード領域とテキスト管理ワークが確保され、コマンドの上からもVZを起動できます。ただしこの場合も新たにテキストをオープンする場合には、EMSメモリが必要となります。

　実は、「**SW**3」で常駐時でも「symdeb」の上でVZが起動してしまいます。これは、「symdeb」が、起動後にデバッグ用に自身のメモリを解放しているためです。この状態では、デバッグ用のエリアにVZがロードされますので、このエリアは使用できなくなります。

第8章 他機種版・付属ユーティリティ

8.1 マルチ機種対応

8.2 J-3100版

8.3 IBM PC／AX版

8.4 EZKEY

8.5 VMAP

8.1 マルチ機種対応

Ver1.5のVZは、対応機種が次のように広がりました。

PC-9801版	vz.com vz.def	・PC-9801シリーズ
J-3100版	vzj31.com vzj31.def	・J-3100日本語モード
IBM PC/AX版	vzibm.com vzibm.def vzax.def	・IBM PCおよび互換機 ・J-3100英語モード ・AXマシン日本語モード

マスターディスクでは、J-3100版はサブディレクトリ「J31」に、IBM PC/AX版はサブディレクトリ「IBM」にまとめてあります。

8.2 J-3100版

VZのJ-3100版（以下VZJ31）は、次のような特長をもっています。

- ハードウェアスクロールによる超高速スクロール
- 高速画面書き込みルーチン
- シフトモードに対応したファンクションキー表示
- 見やすいボールド（太字）フォント表示

■ FROLL.COM

VZJ31の性能をフルに発揮するには、あらかじめ「froll.com」を次のように常駐させてください。

```
A>froll [a]
```

オプション「a」をつけると、FROLLの機能をVZ以外のアプリケーションでも利用できます。オプションをつけない時はVZの中でしか機能しません。
FROLLは、本来そうあるべきJ-3100の日本語BIOSの機能のいくつかを補います。それは次のような機能です。

- スクロールの高速化
- カーソルの全角文字対応

FROLLを常駐しなくてもVZJ31は動作しますが、画面分割した状態や、上方向のスクロールがたいへん遅くなります。

■ 日本語モードと英語モード

J-3100シリーズは、日本語モードと英語モードの2つのモードをもっています。このうち英語モードは、IBM PC互換となっています。VZJ31は、J-3100の日本語モードでしか動作しません。これは、両モードでVRAMの構成が全く異なるためです。英語モードでは、IBM PC版のVZをお使いください。
2つのタイプのVZを扱うのが紛らわしい場合は、付属の「vz.bat」と「vzsel.com」により、現在のモードを識別して、対応する機種版のVZを起動することができます。

■ ハードウェアスクロール

J-3100の日本語モードではテキストVRAMが存在せず、文字はすべてグラフィックVRAMに展開して表示しています。このため、必然的に他の機種版に比べて表示速度が遅くなります。しかし、VZJ31ではできうる限り高速化に工夫を凝らし、特にスクロー

ルは、VRAMの表示アドレスを変える「ハードウェアスクロール」を用いています。
ハードウェアスクロールは確かに高速ですが、そのために画面の両端が幾分ちらつきます。これは故障ではありません。

■ キー配列の相違

J-3100シリーズはIBM-PC準拠のキー配列を採用していますから、PC-9801とは多少異なります。従って、DEFファイルで記述するキーのいくつかも違う名称になります。

PC-9801	[RLDN]	[RLUP]	[CLR]	[HELP]
J-3100, IBM PC	[PGUP]	[PGDN]	[HOME]	[END]

PC-9801で[**GRPH**]キーを表す「@」は、[**Alt**]キーになります。

これらの違いの他、次のようなキーの組み合わせは、入力できません。

[**Ctrl**]+ [**ESC**], [**TAB**], [**BS**], [**INS**], [**DEL**]

指定の際に注意してください。

■ ファンクションキー表示

J-3100のDOSはファンクションキーの表示を行ないません。従って、VZJ31でファンクションキーを表示しています。
PC-9801版では、DEFファイルのファンクションキー定義は2行でしたが、J-3100版のDEFファイルには4行まで記述できます。各々のシフトキー([**Shift**], [**Ctrl**], [**Alt**])を押している間だけ表示が切り変わります。2行目以降を記述しなければ、対応するシフトキーを押しても、表示は変わりません。

```
* F ファンクションキー表示

"ﾌｧｲﾙ",  ...    ← [Fn]
"ﾓｰﾄﾞ",  ...    ← [Shift]+[Fn]
"CTRL+", ...    ← [Ctrl]+[Fn]
"ALT+",  ...    ← [Alt]+[Fn]
```

ファンクションキーを表示しない場合は、「オプション***Ap***」を「0」に設定してください。
【 ***Ap***$70 :ファンクションキー 】

■ 表示属性

VZJ31の、表示属性（オプションA?）の内容は、次のとおりです。

0	非表示
1 ～ 5, 7	通常文字
6	ボールド
8	アンダーラインのみ
9 ～ 13, 15	通常文字、アンラーライン
14	ボールド、アンダーライン
$70	通常文字、反転
$76	ボールド、反転

■ FEPの制御

VZJ31は、ATOK5／6／7Rを制御できます。その他のFEP（VJE β、松茸等）はサポートしていません。

★注意★

ATOK7Rを使用する場合は、必ず「7to5.com」を常駐してください。

■ 動作しないコマンド・機能

PC-9801版がもつ次の機能は、動作しません。

- ［CTRL］+［テンキー］、［GRPH］+［テンキー］による全角文字入力機能
- 20行モード

8.3 IBM PC／AX版

VZのIBM PC／AX版（以下VZIBM）は、次のような特長をもっています。

- AX日本語／英語両モードで動作
- EGA43行／VGA50行モード対応
- 文字色・背景色とも16色対応
- 12ファンクションキー表示モード

■ AX日本語モード

VZIBMは、そのままAXの日本語モードで使用できます。ただし、DEFファイルは「vzax.def」をお使いください。J-3100版と同様に、「vz.bat」と「vzsel.com」を使用すると、現在のモードを識別して対応するDEFファイルを組み込むことができます。

日本語表示を可能にした結果、英語モードで使用する場合に「80h〜9Fh, E0h〜FCh」にあたる文字は、正常に表示できません。偶然にもIBM罫線キャラクタは、半角カナのエリアにあたるため、正常に表示できます。

また、両モードで動作させるために、VZIBM内のメッセージ類は、すべて英語になっています。

FEPの自動制御はサポートしていません。

■ キー配列の相違

キー配列の相違は、J-3100版と同様です。ただし、次のキーも入力できません。

[**Ctrl**]+ [↑], [↓]

■ ファンクションキー表示

ファンクションキー表示も、J-3100版と同様です。さらに、IBM PC／AT以降の拡張キーボードに合わせて、12個のファンクションキーを表示できます。「オプション ***FK***」を「+」に設定して下さい。(AX版はデフォルトで「+」です)

【 ***FK***- :function key labels 10(-)/12(+) 】

■ 表示属性

IBM PC版では、カラーとモノクロの2つの表示属性が、DEFファイルに記述してあります。デフォルトはカラーモードになっていますので、お使いのディスプレイがモノクロの場合は、次のようにDEFファイルを変更してください。

(カラーモード)

```
;*** for Color Monitor ***
An ...
;* 0

*** for Monochrome Monitor ***
An ...
* 0
```

(モノクロモード)

```
*** for Color Monitor ***
An ...
* 0

;*** for Monochrome Monitor ***
An ...
;* 0
```

表示属性は、上位4bitで背景色、下位4bitで文字色を指定します。どのような属性になるかは、[**ESC**]@コマンドで実際に属性（例えば***An***）を変更してお試しください。

8.4 EZKEY

EZKEYは、メモリに常駐してキー入力割込み（int09h）をハンドルし、次のような機能をサポートするツールです。

- [CTRL]+[GRPH]+[..]によるホットキー機能
- ブルーバック／ホワイトバック機能（アナログRGB使用）
- [CAPS], [ｶﾅ]キー表示
- [SHIFT]押下時の裏ファンクションキー表示 ※
- VZ上での[GRPH], [NFER], [CTRL]+[f･n]キーのサポート ※

※ エプソンDOS、エプソン98互換機では、一部の機能が動作しません。

EZKEYは、PC-9801ノーマルモードでのみ使用できます。

■ EZKEYの組み込み

EZKEYは、次のようにメモリに常駐してお使いください。

```
A>ezkey {-&lt;オプション&gt;[+|-]}
```

[例]

```
A>ezkey -b3 -c -f-
```

取り外す時は、再度

```
A>ezkey
```

を実行します。また、

```
A>ezkey -?
```

で、<オプション>の一覧が表示されます。

ダイレクトコマンド

EZKEYを常駐すると、[CTRL]+[GRPH]+[英字]を押すことで以下のダイレクトコマンドを実行します。これらの機能は、対応するオプションを「-」で指定することで禁止できます。

[例]

```
A>ezkey -i-
```

[英字] 機能	-<オプション>
[G] clear Graphic vram	-g

グラフィックVRAM（バンク0）をクリアします。E0000h〜E8000hの16色表示用VRAMも含みます。

[T] clear Text vram	-t

テキストVRAM（ページ0）をクリアします。文字の属性は、DOSの既定値で初期化します。

[S] Save text vram	-s

表（ページ0）テキストVRAMの内容を、裏（ページ1）テキストVRAMへ、コピーします。後で参照したい画面を保存するときに使います。

[X] eXchange text vram	-x

画面に表示するテキストVRAMを、表（ページ0）または裏（ページ1）とで切り替えます。[S]で保存した画面を参照するときに使います。

[L] Lock keyboard	-l

[CTRL]+[GRPH]+[L]を押すと、ビープ音が短く鳴り、キーボードをロックします。再度[CTRL]+[GRPH]+[L]を押すと、ロックは解除されます。

[DEL] reset cpu	-!

リセットします。286／386機や、EMSボードを搭載している場合は、正常に動作しない場合があります。

[I] Interrupt　　　　　　　　-i

INT3を発生し、symdeb等の上で実行中のプログラムをブレークします。通常は使用してはなりません。

モードコマンド

EZKEY常駐時に、次のモードをオプションで指定できます。モードは[**CTRL**]+[**GPRH**]+[英字]キーで切り替えられます。

[英字]　機能　　　　　　　　-<オプション>

[**D**] Display graphic vram　　-i-

グラフィックVRAMの表示／非表示を切り替えます。

[**A**] Analog/digital mode　　-a-

グラフィックのデジタル／アナログRGBモードを切り替えます。

[**B**] Blue back　　　　　　-b- (4)

アナログRGBで、ブルーの背景色を表示します。「0~7」で明るさを指定します。(デフォルト4) また、[**CTRL**]+[**GRPH**]+[**B**]で、表示のON／OFFを切り替えられます。

[例]

```
A>ezkey -b      (明るさは4)
A>ezkey -b7     (明るさは7)
```

[**W**] White back　　　　　　-w- (4)

アナログRGBで、グレーの背景色を表示します。それ以外は [**B**] と同じです。この時、VZを -an0 オプションで起動すると、白地に黒文字表示となります。

[R] type caps for RA, RX -r5 ※

[CAPS]または[ｶﾅ]キーがロックされていると、画面最下行のファンクションキーの右端に「CAPS」または「ｶﾅ」を表示します。数値は「1〜7」で、表示色を指定します。デフォルトは5（水色）です。

[例]

A>ezkey -r6　（黄色で表示）
A>ezkey -r-　（[CAPS], [ｶﾅ]の表示をしない）

[C] Click sound -c- (4)

キー入力時にクリック音を発生します。「0〜7」で、音の長さを指定します。標準設定は4です。

[F] shift Function key -f+

[SHIFT]キーを押している間、裏ファンクションキー（f•11〜f•20）の内容を表示します。本機能を使うと、DOSのファンクションキー表示を使用していないソフトが、正規の方法でファンクションキーを消していない場合、表示が乱れる場合があります。そのようなときは、[CTRL]+[SHIFT]+[F] でこの機能を禁止してください。

[¥] character -¥-

[SHIFT]+[^]でバッククォート文字 [`] を入力します。
" " " " " " " " " "

[@] activate [GRPH] key-@+ ※

VZ上で、[GRPH]+[英字]コマンド、および [GRPH]+[テンキー]による全角記号の入力を可能にします。

[**N**] activate [NFER] key　　　-N+　　※

VZ上で、[**NFER**]キーを使用可能にします。

[^] activate [CTRL]+[f•n] key　-^-　　※

VZ上で、[**CTRL**]+[**f•n**]キーを使用可能にします。

※ VJE β、松茸等のFEPで漢字入力モードがONになっている時は、動作しません。

8.5 VMAP

VMAPは、メインメモリおよびEMSの使用状況を表示するMS-DOS汎用ツールです。これを利用することで、メモリの残り容量、常駐（TSR）プログラムのサイズ等を知ることができます。ちなみに「VMAP」は、「View memory MAP」の略です。

VMAPは、通常のコマンドとして実行します。オプション指定はありません。

■ 実行例

```
A>vmap
VMAP Version 1.0x  Copyright (C) 1989 by c.mos

addr PSP  blks   size  owner/parameters           hooked vectors
---- ---- ---- ------  ------------------------   ------------------
0D52 0008   1   50896  config                     0C 19 1B 67 DC
19C0 19C0   2    4464  shell                      22 23 24 2E
1AD9        1      48  <free>
1ADD 1AF1   2     544  autoswch
1B01 25FD   1     304  vz (env)
1B15 1B15   1    1472  ezkey                      09 E0
1B72 1B72   1   43168  CON                        70
25FD 25FD   1    2432  vz -sw3 -z              06 21 29
2696        1  497296  <free>

handle page   size  name     (EMS frame:C000h)
------ ---- ------  --------
     1  100  1600k  IOS10
     2   27   432k  VZ
  free     0    0k
 total  127  2032k
```

※　各数値は、実際のものとは異なります。

■ 各シンボルの説明

addr	MCB (Memory Control Block) の開始セグメント
PSP	PSP (Program Segment Prefix) のセグメント
blks	MCBのブロック数（連続している場合に限り加算される）
size	MCBのバイト数
owner	TSRのコマンド名
parameters	実行時のパラメータ
hooked vectors	TSRがフックする割り込み番号
config	config.sysで組み込んだデバイス
shell	command.comの常駐部
<free>	空き領域
auto..,vz,ezkey	常駐プログラム
.. (env)	環境領域
CON	adddrvで組み込んだデバイス名
handle	EMSハンドル番号
page	使用ページ数
size	使用ページの合計Kバイト数
name	ハンドル名（EMM4.0使用時）
frame:	フレーム開始セグメント

第9章　テクニカルノート

9.1 モジュールの概要
9.2 アセンブル・リンク
9.3 割り込みハンドル
9.4 FEPの制御
9.5 グラフキーの利用

9.1 モジュールの概要

この章では、ソースコードをアセンブル・リンクして各機種版のVZを作成する方法の他、プログラミングに関する幾つかの留意点についてコメントします。本章をお読みになる際は、アセンブラプログラミングに関しての多少の知識が必要です。

VZエディタVer1.5アセンブラソースコードの各モジュールの概要を、以下に示します。

ファイル名		内容
std.inc		標準のインクルードファイル
vz.inc		VZのインクルードファイル
main.asm		イニシャライズ
alias.asm		エイリアス処理
char.asm		文字列処理サブルーチン
core.asm		編集モードの基本ルーチン
disp.asm		各種表示処理
dos.asm		DOSコマンドライン、常駐処理
expr.asm		オプション、マクロ演算式の処理
filer.asm		ファイラー
gets.asm		文字列入力、ヒストリー処理
harderr.asm		致命的エラーの処理ルーチン
inst.asm		カスタマイズ
key.asm		キー入力処理（共通部）
key98.asm	*	キー入力処理（PC-9801）
keyibm.asm	*	キー入力処理（J-3100, IBM PC）
ledit.asm		ラインエディット
macro.asm		マクロ機能
memo.asm		ブロックコマンド、メモリ管理
menu.asm		メニュー関連
misc.asm		その他の編集モードコマンド
open.asm		ファイル入出力関連コマンド
printf.asm		printfルーチン
scrn.asm		画面出力処理（共通部）
scrn98.asm	*	画面出力処理（PC-9801）
scrnibm.asm	*	画面出力処理（J-3100, IBM PC）

smooth.asm		スムーススクロール
string.asm		文字列の検索・置換等
text.asm		ファイルアクセス、テンポラリ管理
view.asm		カーソル移動コマンド
wind.asm		ウィンドウ・マルチファイル管理
xscr.asm		コンソール出力の取り込み
swap.asm		スワップ処理
ems.asm		EMSハンドラ
msg.asm	*	各種メッセージ類

* は機種依存モジュール

9.2 アセンブル・リンク

VZのソースコードをアセンブル・リンクする時は、次のバッチファイル、MAKEファイルをお使いください。

ファイル名	内容
vz.lnk	リンクファイル
vz.mak	標準MAKEファイル
vz.omk	OPTASM用MAKEファイル
ac.bat	アセンブル用バッチファイル
lk.bat	リンク用バッチファイル
mk.bat	MAKE用バッチファイル

■ アセンブラ

標準添付の「vz.com」は、OPTASM Ver1.6でアセンブルされていますが、次のアセンブラでもアセンブルできます。

```
OPTASM 1.5  MASM 4.0/5.0/5.1  TASM 1.0
```

ただし、OPTASM以外のアセンブラを使用すると、作成した「vz.com」のサイズが多少大きくなる場合があります。

使用するアセンブラに応じて、あらかじめ次の環境変数を設定してください。

```
OPTASM: A>set asm=opt
MASM:   A>set asm=m
TASM:   A>set asm=t
```

「ac.bat」で1つのファイルをアセンブルできます。

```
A>ac main
A>ac main l
```

2つ目の例では、アセンブルオプション「/l」を指定したことになります。

■ リンカ

リンカとして次のものが使用できます。

```
TLINK  MS-LINK
```

TLINKを使用する場合は、次の環境変数を設定します。

```
TLINK:  A>set lnk=t
```

※ MS-LINKを使用する場合は、他に「exe2bin.exe」が必要です。

「lk.bat」でリンクのみ実行できます。

```
A>lk
```

このバッチファイルでは、リンクファイル「vz.lnk」を参照しています。

■ MAKEユーティリティ

VZのソースコードを一括してアセンブル・リンクするために、MAKEユーティリティを利用すると便利です。ただしOPTASMはMAKE機能を内蔵していますので、MAKEは必要ありません。

次のMAKEユーティリティが使用できます。

```
MS-MAKE 4.0  Turbo-MAKE 2.0
```

VZは次の方法で作成できます。

```
A>mk [98|j31|ibm]
```

機種指定を省略した場合は、最後に指定した機種版を作成します。

■ アセンブルスイッチ

VZのソースコードの中には、いくつかのアセンブルスイッチが使用されています。これらを設定することで、ソースコードに手を加えることなく、各種タイプのVZを生成します。

スイッチ	機能
PC98	PC-9801版（デフォルト）
J31	J-3100版
IBM	IBM PC／AX版
V30	186命令使用
MINI	スモール版
NOFILER	ファイラーを除く
NOALIAS	エイリアス処理を除く
NOXSCR	コンソール画面取り込み処理を除く
NOSWAP	スワップ機能を除く

9.3 割り込みハンドル

添付のコマンドは、次の割り込みベクタをフックします。

割り込み番号	機能
< VZ >	
06h	[STOP]キー割り込み（PC-9801）
1Bh	[Ctrl]+[Break]キー割り込み（J-3100, IBM PC）
21h	システムコール（常駐時のみ） （ファンクション0Ah、4Bhをハンドル）
24h	致命的エラーの処理（VZ動作中のみ）
29h	コンソール1文字出力
< EZKEY >	
09h	キー入力割り込み（PC-9801）
< FROLL >	
10h	CRT　BIOS（J-3100）

9.4 FEPの制御

VZでは、入力モードに応じて、日本語入力フロントエンドプロセッサ（FEP）を自動ON／OFFする機能をもっています。対応するFEPは、VJE β（V2.0)、松茸（V2.0)、ATOK6／7です。

■ FEPの自動識別

それぞれの割り込みベクタからの指定オフセットに、指定の文字列が格納されているかどうかで識別しています。

FEP	割り込み	オフセット	文字列
VJE β	70h	12h	"VJ"
松茸	70h	12h	"MT"
ATOK6	6Fh	ベクタ＋03h	"AT"
ATOK7	6Fh	ベクタ＋33h	"K7"

■ FEPの制御

FEPのON／OFF制御は、次のロジックで行ないます。

- 編集モードから他のモード（ファイル名入力、コマンドライン等※）に移る時、その時のFEPの状態を取得・保存して、FEPをOFFする。
- 他のモードから編集モードに戻る時、保存した状態を復帰する。

さらに、「オプションFR」により

- 起動時、編集モードに入ると同時にFEPをONする。

※　検索・複写文字列の入力時には、FEPを制御しません。

メニュー選択、(Y/N)入力、ファイラーモード等の1文字入力時には、BIOSから直接キー入力することで、FEPの変換は避けられます。ただし、ATOKはBIOS入力にも変換を被せていますので、「ファンクション05」をコールすることで、これをも回避しています。

しかし、J-3100版のATOKでは、この方法で回避することができません。ATOK5／6では、「一時的マスク」ファンクションが使えます。しかし、ATOK7はこのファンクションをサポートしておらず、なおかつ付属の「7to5.com」はこの部分が正常に動作しません。ゆえに苦肉の策として、ATOKがフックした割り込みベクタの元のアドレスを検出し、一時的に割り込みベクタを元に戻してからコールしています。

■　その他のFEPの対応

VJE、松茸、ATOK以外のFEPを自動ON／OFFする場合には、そのFEPの制御ルーチンを、ソースコードのkey???.asm に記述して再アセンブルしてください。この時必要となるFEPの制御ルーチンは、次のとおりです。

(1) FEPの識別（上記のような割り込みベクタ番号と識別文字）
(2) FEPのON
(3) FEPのOFF
(4) FEPの現在のモードの取得
(5) FEPの一時的マスク（BIOS変換を被せる時）
(6) FEPの一時的マスク解除

9.5 グラフキーの利用

PC-9801では、MS-DOSからのキー入力で、[GRPH]+[英字], [NFER], [CTRL]+[f･n]キー等を取得することはできません。VZでは、EZKEYを介して次のような方法でこれらのキーを取得し、コマンドキーとして使用可能にしています。

まず、これらのキーが押されると、キー入力割り込みルーチン内で、EZKEYがキーコードをDOSから入力可能なコードに変換します。

[GRPH]+[英字]　→ [CTRL]+[英字]
[NFER]　→ [CTRL]+[_]
[CTRL]+[f･n]　→ [f･n]

そして、これらのキーをVZのキー入力ルーチンで受け取った時、BIOSの「キーの押下情報テーブル」を参照し、実際に押されたキーが何であるかを判定して、正しいキーコードに変換します。

[GRPH]キーを他のソフトで使用している場合には、EZKEYが変換してしまうと、そのソフトでは正常に入力されなくなります。これを避けるために、VZは、EZKEYが常駐しているかどうか調べ、常駐している場合にはVZの編集モード時のみ、[GRPH]キー等の変換を行なうように、EZKEYを制御しています。したがって、VZを使用していない時には、EZKEYを常駐させていても、[GRPH]キー等の変換は行なわれません。

しかし、FEPが[GRPH], [NFER]キーを使用する場合には、そのように切り分けることができません。EZKEYでは、この問題に対処するために、FEPの変換モードがONかOFFかを判断して、FEPがONになっている場合には、[GRPH], [NFER]キー等の変換を抑止しています。これは、次のBIOSワークを参照しておこなっています。

0:053Dh　01　: FEP　OFF
　　　　　02　: FEP　ON

実は、このワークは、テキストVRAMのウィンドウ分割数を表しており、FEPの漢字入力モードがONすると、画面最下行に別のウィンドウを表示することを利用しています。従って、ATOKなど、ウィンドウを使用しないFEP、あるいは、ウィンドウを使用しない設定でFEPを使用した場合には、正常に判定できません。このような状況化で、[GRPH], [NFER]キーをFEPで使用する場合は、

```
A>ezkey -@- -N-
```

としてください。

第10章 VZのカスタマイズ

10.1 カスタマイズの概要

10.2 DEFファイルの分割

10.3 DEFファイルの取り込み

10.4 編集時の再カスタマイズ

10.5 DEFファイルの書式

10.6 起動時オプション

10.1 カスタマイズの概要

「カスタマイズ(customize)」とは、アプリケーションの使用環境を、ユーザーが各自の好みにあわせて再設定することです。VZでは、次のような種々の環境をカスタマイズできます。

- コマンドキーの割り当て
- ファンクションキーの表示ラベル
- ウィンドウのタイトル
- ポップアップメニュー
- マクロコマンド
- オプション(メモリ・バッファサイズ、表示属性、etc.)

これらの情報は、「DEFファイル」と呼ばれるテキストファイル(拡張子.def)に記述してあります。標準のDEFファイルは、「vz.def」ですが、この他にも数種類のDEFファイルが用意されています。

<PC-9801版>	
vz.def	VZ標準DEFファイル
vze.def	VZ標準DEFファイル(英語表記)
vzs.def	スモール版DEFファイル
ws.def	WordStarライク
mi.def	Mifesライク
<J-3100版>	
vzj31.def	VZJ31標準DEFファイル
wsj31.def	WordStarライク
<IBM PC/AX版>	
vzibm.def	VZIBM標準DEFファイル(英語表記)
vzax.def	AX用標準DEFファイル
wsibm.def	WordStarライク(英語表記)

標準DEFファイル（起動コマンド名.def）以外のDEFファイルを使用する場合は、次の2通りの方法で指定してください。

■ 起動時に、コマンドラインで指定する。

［例］

```
A>vz /ws
```

■ 環境変数「VZDEF」に設定する。

［例］

```
A>set vzdef=ws
```

10.2 DEFファイルの分割

DEFファイルは、複数に分割することができます。そして、次の3つの方法のいずれかで組み込みます。

(1) 起動時に読み込む
(2) 「vz.com」本体に組み込む
(3) 編集中に読み込む

(2)については『10.3 DEFファイルの取り込み』で、(3)については『10.4 編集時の再カスタマイズ』のところで説明しますが、こうすることにより次のようなことが可能になります。

・DEFファイルを固定部分と可変部分（マクロ、エイリアス、ヒストリー等）に分割し、固定部分は「vz.com」本体に組み込み、可変部分のみ起動時に読み込む。
・マクロ定義を機能ごとに別ファイルとし、必要に応じて組み込む。

起動時に複数のDEFファイルを読み込む場合は、最初に読み込むDEFファイルの「*E その他 1」の項目に、2番目以降のDEFファイル名を記述します。複数のファイルは「+」でつなげて指定します。間に<ｽﾍﾟｰｽ>を入れてはいけません。

[例] 「vzs.def」ファイル

```
* E その他

1       help+Update
```

※ Updateが大文字で始まっている理由は次節で説明します。

「vzs.def」ファイルは、スモール版VZ「vzs.com」のDEFファイルですが、サイズを小さくする意味で、ヘルプ、エイリアス定義等は含まれていません。これらを使用する場合は、それぞれ「help.def」、「update.def」を組み込みます。上の［例］のように修正したら、次のように「vzs」を実行してみてください。

```
A>vzs -st

VZ Editor (S) Version 1.5x  Copyright (C) 1989 by c.mos
======== vzs.def (v1.5) ========
======== help.def (v1.5) ========
======== update.def (v1.5) ========
```

と画面に表示され、vzs.def、help.def、update.defファイルが順次読み込まれたことを示します。「オプション***St***」は、起動時に読み込んだDEFファイルの先頭の行をコンソールに出力するオプションです。

【***St***-　:DEFファイルのタイトル表示 】

さらに、起動時のコマンドラインからも読み込むDEFファイルを指定できます。

［例］

```
A>vz +block
```

「+」の後にDEFファイル名を指定してください。複数のDEFファイルを読み込む時には、「+」でつなげて指定します。間に<ｽﾍﾟｰｽ>を入れてはいけません。

［例］

```
A>vz +block+tool
```

以上の仕様にもとづき、読み込まれるDEFファイルの例を示します。

［例］

```
A>vz            → vz
A>vz +block     → vz, block
A>vz /vze       → vze, help
A>vz /vze+block → vze, help, block
A>vz /vze+      → vze
```

※　最後の例のように終わりに「+」を指定すると、「* E その他 1」で記述されたDEFファイルを読み込ませないことができます。

10.3 DEFファイルの取り込み

起動時に「オプション**W**」を指定すると、DEFファイルを「vz.com」の中に取り込むことができます。

```
A>vz -w
VZ Editor Version 1.5x  Copyright (C) 1989 by c.mos
"A:¥VZ.COM"を更新しました.

A>dir vz.com
...
VZ       COM    51623  89-12-02    1:50
※  実際のサイズとは多少異なります。
```

DEFファイルを取り込むと、「vz.com」本体のみでVZが起動できる他、DEFファイルを読み込む時間が要らなくなる分、起動時間が短縮できます。

再び取り去る場合は、

```
A>vz -w-
```

とします。ただし、取り込む前と全く同じ内容になるわけではありません。デフォルトのオプションは、組み込んだDEFファイルの設定がそのまま残ります。

先に述べた「vzs.def」の例で、DEFファイルの固定部分（vzs.def, help.def）のみを取り込み、可変部分（update.def）は後から読み込ませることもできます。

```
A>vz /vzs -w
```

こうすると、「vzs.def、help.def」のみ取り込まれます。これは、「vzs.def」で次のように指定しているためです。

```
* E その他

1       help+Update
```

このように、先頭を「英大文字」で指定したDEFファイルは、「-**w**」で取り込まず、必ず起動時に読み込みます。

10.4 編集時の再カスタマイズ

DEFファイルのマクロやメニューを作成している場合、それをその場で組み込み、実行して動作を確認できると、たいへん便利です。VZではそのためのコマンドも用意しています。

[CTRL]+[K][^]　　　:再カスタマイズ

編集中のテキストをDEFファイルとして組み込みます。コマンドを実行した結果正常に組み込まれると、

《・・項目読み込みました》

と表示されます。この時の「項目」は、メニュー、マクロ、エイリアスの個数に相当します。ファイルにエラーがある場合は、

《カスタマイズエラー》

と表示され、カーソルをエラーの発生した行へ移動します。また、組み込み中にマクロバッファの容量が足りなくなった場合は、

《バッファが足りません》

と表示され、その時読み込んでいた行へカーソルを移動します。このコマンドで新たなマクロ・メニューまたはエイリアスを組み込む場合には、それぞれ「オプション ***Bm***」および「オプション ***Ba***」で、起動時に十分なバッファを確保しておかなければなりません。

【 ***Bm*** 1024 :マクロバッファのフリーエリア 】
【 ***Ba*** 0 　:エイリアスバッファフリーエリア 】

このコマンドで組み込める項目は、DEF ファイルの項目のうち、

---

* M マクロ
* P メニュー
* S システムメニュー
* A エイリアス

に限定され、他の項目は無視します。

外部マクロの実行や、編集中のエイリアスの再定義は、この機能で実現しています。

再カスタマイズで組み込んだものは、以下の要領で既存の定義と置き換わります。

■ マクロ、メニュー

再カスタマイズで組み込むマクロやメニューで最初に定義されているマクロ番号（メニュー番号）を、マクロバッファ内でサーチし、見つかった場合はそれ以降を削除して、新らしく組み込む内容と置き換えます。マクロとメニューは、同じバッファに格納されています。

■ エイリアス

既存の定義は破棄され、新しく組み込む内容と置き換えます。

このため、例えば「vze.def」を組み込み、「vze.def」ファイルのメニュー等を変更して再カスタマイズすると、「help.def」の内容を失なってしまいますので注意が必要です。

10.5 DEFファイルの書式

この節では、DEFファイルの書式について説明します。

まず、DEFファイル全般にわたる事項について触れます。

■ DEFファイルの最初の行は、タイトル行としてスキップします。また、「オプション ***St***」を「+」に指定すると、組み込み時にその行をコンソールに出力します。
【***St***-　　:DEFファイルのタイトル表示 】

■ 「*」で始まる行は、続く1文字の英字で、以下の10種類の項目の開始を表します。

(日本語表記)	(英語表記)
* C コマンドキー	* Command keys
* F ファンクションキー表示	* Function key labels
* T テキストウィンドウ	* Text windows
* S システムメニュー	* System menu
* M マクロ	* Macros
* P メニュー	* Pop up menu
* O オプション	* Options
* A エイリアス	* Alias
* H ヒストリー	* History
* E その他	* Else

■ 「*」のみの行は、DEFファイルの終りを表しています。これ以降の記述は読みとばします。

■ 「;」で始まる行、および「*Alias *Histoy *Else」以外の項目の「;」から行末までは、コメントとして読みとばします。

C コマンドキー

コマンドキーの割り当て、およびそのヘルプメッセージを記述します。

[書式]

```
00  {<prefix key>}   [;<comment>]
<n> [<key1>][<key2>] [:<message>] [;<comment>]
```

<n>	: コマンド番号 (01～99)
<key1>	: コマンドキー定義1
<key2>	: コマンドキー定義2
<prefix key>	: 2ストロークキーの先頭キー (最大6個)
<message>	: ヘルプメッセージ
<comment>	: コメント

コマンドキーとして、次のキーが指定でき、各々別の機能を割り当てられます。

```
^<英字キー>  @<英字キー>  <prefix><英字キー>
 [特殊キー]  ¥[特殊キー]  ^[特殊キー]  @[特殊キー]
```

<PC-9801版>

^	: [**CTRL**]
¥	: [**SHIFT**]
@	: [**GRPH**]
<英字キー>	:@ A..Z [¥] ^ _ (40h～5Fh)
[特殊キー]	: [RLUP] [RLDN] [INS] [DEL] [UP] [←] [→] [DOWN] [CLR] [HELP] [ESC] [TAB] [BS] [↵] [NFER] [F01]～[F15]

<prefix>: ^<英字キー> @<英字キー> [特殊キー]

<J-3100,IBM PC版>

^	: [**Ctrl**]
¥	: [**Shift**]
@	: [**Alt**]
<英字キー>	:@ A..Z [¥] ^ _ (40h～5Fh)
[特殊キー]	: [PGDN] [PGUP] [INS] [DEL] [UP] [←] [→] [DOWN] [HOME] [END] [ESC] [TAB] [BS] [↵] [F01]～[F12]

<prefix>: ^<英字キー> @<英字キー> [特殊キー]

■ 制限事項

(A) [特殊キー]と、3番目以降の〈prefix〉キーは、各々1つしか指定できません。

(B) 3番目以降の〈prefix〉キーは、先に指定しなければなりません。

(C) 01〜08番のコマンドには、〈prefix〉〈英字キー〉は指定できません。

(D) PC-9801版の@〈英字キー〉、[NFER]は、EZKEYを組み込まないと使用できません。

(E) 同じく^[F01]〜^[F10]は、EZKEYを「-^」オプションで組み込まないと使用できません。

[誤った記述例]

11	[HOME] ^[←]	:論理行頭	★制限(A)
70	@0 [ESC]0	:0 ファイ...	★制限(B)
02	^KM	:キーボー...	★制限(C)

[Ver1.5で可能になった記述例]

46	^B ^KB	:Mark Block
70	[ESC]0 @0	:0 ファイルのオープン

■ 入力できないキー

機種によって、次のキーの組み合わせは入力できません。

＜PC-9801版＞

^[CLR] @[CLR] @[F01]〜@[F15] @[@] @_

＜J-3100版＞

^[ESC] ^[TAB] ^[BS] ^[INS] ^[DEL]
@[特殊キー] @〈英記号〉

＜IBM PC版＞

^[ESC] ^[TAB] ^[BS] ^[INS] ^[DEL] ^[UP] ^[DOWN]
@[特殊キー] @〈英記号〉

■ メモリ消費

コマンドキー定義は固定サイズのテーブルへ書き込みますので、定義の内容により消費メモリが変わることはありません。しかし、:<message>で記述したヘルプメッセージはそのまま格納しますので、あまり冗長なメッセージを記述すると、メモリの無駄使いとなります。

F ファンクションキー表示

ファンクションキーの表示ラベルを記述します。

[書式]

```
"<f･1>","<f･2>", ... ,"<f･10>"  (PC-9801)
"<f･1>","<f･2>", ... ,"<f･12>"  (IBM PC)
```

<f･n>　　:[f･n]キーの表示ラベル
最大5文字（PC-9801版,IBM PC）
最大6文字（J-3100版）

T テキストウィンドウ

文字列入力ウィンドウのタイトルメッセージ、サイズを記述します。

[書式]

```
"<title>",<width> [;<comment>]
```

<title>　　:ウィンドウのタイトルメッセージ
<width>　　:ウィンドウの横サイズ
<comment>　　:コメント

システムで使用する10種類の文字列入力ウィンドウを順次記述します。

●● 文字列内の制御文字 ●●

「$」は、文字列中の制御文字で、次のような意味をもちます。

$"	'"'
$$	'$'
$0〜$7	表示色
$!	リバース
$a?	「オプションA?」の表示色（?は英字）
$an	テキストの表示色
$p?	「変数P?」の示す文字列
$pc	コマンド名
$pn	カレントドライブ名
$pd	カレントディレクトリ
$??	「変数??」の10進表示
$lc	表示行数
$ht	編集テキストのタブカラム数
$(hh, ee, xx, ..)	16進コード指定（hh, ee, xxは16進数）

［例］

```
66      ^KL               :L 表示行数切替え ($lc)
"【入力ファイル】 <$pd>",40
15 ^KI "T タブの切替え   ($ht)"
" $4EMS free/total: $ef/$em"
8       $6$pd>
```

P メニュー / S システムメニュー

マクロメニューとシステムメニューを記述します。

[書式]

```
<n> "[<title>]",<width>,<items>[,<value width>][,-1] [;<comment>]
    { ["<item>"] [,#nn|&nn|!nn|<option>] }
```

<n>	：メニュー番号（システム：1～9、マクロ：1～98）
<title>	：メニューのタイトルメッセージ
<width>	：メニューの横サイズ
<items>	：メニューアイテムの個数
<value width>	：パラメータのカラム数
<comment>	：コメント
<item>	：アイテムメッセージ
#nn	：コマンド番号（nnは2桁の10進数）
&nn	：マクロ番号
!nn	：メニュー番号
<option>	：オプション

■ システムメニュー

システムメニューは、コマンドまたはファイラーから使用されるメニューで、必ず定義する必要があります。また、5,8,9番以外のシステムメニューのメニューアイテムの個数を変更してはいけません。

システムメニューは、マクロ・マクロメニューに先んじて定義しなければなりません。

■ マクロメニュー

マクロメニューは、マクロ中で使用するメニューで、自由に定義して使用できます。定義したメニューは、マクロまたは他のメニューから、「!nn」で呼び出すことができます。

■ マクロメニューのタイトル

マクロメニューのタイトルメッセージ（<title>）を省略した場合は、

・マクロから呼び出された場合 ………. マクロのタイトルメッセージ

・他のメニューから呼び出された場合 …. 呼び出したアイテムメッセージ

がタイトルメッセージとなります。

メニューの各アイテムメッセージの後に以下のような記述をすると、そのメニューを選択して各種機能を実行できます。

■ コマンド番号（#nn）

nn番のコマンドを実行します。アイテムメッセージ（<item>）を省略すると、「* C コマンドキー」の :<message>で記述したヘルプメッセージを表示します。

■ マクロ番号（&nn）

nn番のマクロをコールします。アイテムメッセージ（<item>）を省略すると、コールするマクロのタイトルメッセージを表示します。

■ メニュー番号（!nn）

nn 番のメニューを表示します。

■ オプション（<option>）

指定オプションがスイッチオプションならスイッチのON／OFF、その他のオプションなら、その値を入力します。

通常、スイッチの状態は『ON』、『OFF』で表示されますが、その他のメッセージを表示したい場合は、アイテムメッセージ中に、次のような書式でスイッチメッセージを記述してください。

[書式]

"<アイテムメッセージ><ﾀﾌﾞ><OFFメッセージ><ｽﾍﾟｰｽ><ONメッセージ>"

[例]

"N 行番号モード<ﾀﾌﾞ>論理 表示",Dn

※ 具体例は、各種DEFファイルを参考にしてください。

M マクロ

マクロプログラムを記述します。

[書式]

```
<n> <key> "<message>" <statement> (グローバルマクロ)
<n>: <statement>             (ローカルマクロ)
```

<n>　　　　　　：マクロ番号（1～89）
<key>　　　　　：マクロキー
<message>　　：マクロのヘルプメッセージ
<statement>　：マクロ定義の本体

ローカルマクロは、他のマクロからコールされる、いわばサブルーチンです。<statement>はマクロ定義の本体で、行頭が「数字」または「*」の行までを、一つのプログラムと認識します。メモリに格納する際にスペース、タブ、改行、コメントは取り払われます。マクロに関する詳しい説明は、次章でおこないます。

O オプション

起動時オプションを記述します。
[書式]

```
<option>[n|+|-] [;<comment>]
```

<option>　　：オプション
<comment>　：コメント

全オプションの機能については、次節で説明します。

A エイリアス

コマンドラインで使用するエイリアス定義を記述します。
[書式]

```
<atom><ｽﾍﾟｰｽ><command>
```

<atom>　　　：エイリアス名
<command>　：コマンド

エイリアス機能については『6.2 エイリアス』をご覧ください。

H ヒストリー

ヒストリーバッファの初期文字列を記述します。
[書式]

```
:string | :file | :command | :application | :work
{ <word> | "...." }
```

行頭の「:」に続く一文字の英字（s|f|c|a|w）で、5種類のヒストリーバッファを選択し、その後に文字列を記述します。

string	：検索・置換・複写文字列バッファ
file	：ファイル名バッファ
command	：コマンドラインバッファ
application	：アプリケーションコマンドバッファ
work	：その他のバッファ（^QJ ［ESC］@等で使用）

各文字列はスペースまたは改行で区切ります。文字列中にスペースを指定したい場合は、文字列を「"」でくくってください。

ヒストリー機能については、『4.3 ヒストリー機能』をご覧ください。

E その他

その他の10種類の環境文字列を記述します。
[書式]

```
<1～10> <string>
```

<string>の後にコメントを加えることはできません。

各項目の機能は、次のとおりです。

1	help+Update	2番目以降に読み込むDEFファイル 『10.1 カスタマイズの概要』
2	editfile	プロファイル機能で参照するファイル名 『3.3 プロファイル機能』
3	.doc .txt .bat ...	デフォルト拡張子 『2.3 編集ファイル名の省略』
4	Drive Mask ...	ファイラーメニュー 『5.3 メニューコマンド』
5	0●,◎.○1《 ...	[GRPH]+<テンキー> 『4.7 特殊文字の入力』
6	0└,┴.┘1├ ...	[CTRL]+<テンキー>による 『4.7 特殊文字の入力』
7	*/;^L	タイトル行サーチのサーチ文字 『4.1 カーソルの移動』
8	6pd>	非常駐時のコマンドラインプロンプト 『6.1 コマンドラインモード』
9	.com .exe .obj ...	デフォルトバイナリー拡張子 『2.3 編集ファイル名の省略』
10	"$(95,96,9C,9D, ...	ウィンドウ枠、CR,TAB,EOF文字コード

10.6 起動時オプション

VZは、たいへん多くのオプションを備えています。全ての指定可能なオプションは、DEFファイルの「* 0 オプション」の箇所で定義してあります。また、起動時のコマンドラインで「-」をつけて指定することもできます。

■ オプションは、次のように分類して説明します。

・動作モード指定
・メモリ／バッファサイズ指定
・各種モード指定
・表示属性
・スイッチ

■ 書式上の注意

・オプションでは英大文字と小文字は識別しません。
・オプションに続く( )の値はデフォルト値です。
・オプションの後の[ ]は単位です。
・オプションの後に「*」がついているものは、実行中に変更可能です。
・オプションの後の「1.5」は、Ver1.5で追加／変更されたものです。
・スイッチの『……』は、それを変更するモードメニューです。
・スイッチの説明は、特に触れてない限り「+」の場合の効果です。

■ 動作モード指定

z	VZを常駐または解放します。
w[-]	DEFファイルを「vz.com」に取り込み／取り外します。
r[<n>]	起動後、ただちに番号で指定したマクロを実行します。番号を省略した場合は、ファイルで最初に現れたマクロを実行します。

■ メモリ／バッファサイズ指定

Bt(64) [KB] テキストバッファのサイズを設定します。大きなファイルを複数個編集する場合は、この値を小さくしてください。
[範囲] 16 ～ 64

Bq(0) [KB] 指定するサイズまで、テンポラリファイルをメモリ上に作成します。RAMディスク等がない場合にはこれを設定します。

Bf(32) [KB] DOSコマンド用に確保する最低のメモリサイズを設定します。通常VZが使用していないメモリは、DOSコマンド実行時に解放されます。EMSを使用できる場合は、「Bf1」に設定することで、テキストバッファをEMS上に確保します。

Bo(32) [KB] コンソール出力バッファのサイズを設定します。「Bo0」でこの機能は使えなくなります。
[範囲] 0 ～ 32

Bm(256)[byte] マクロバッファの空き領域のサイズを設定します。この領域は、キーボードマクロ、および外部マクロ読み込み用バッファとして使用されます。
[範囲] 256 ～

Ba(0) [byte] エイリアスバッファの空き領域のサイズを設定します。実行時にエイリアスを再定義する場合には、この値を増やしてください。

Hs(256)[byte] 検索文字列ヒストリーバッファの空き領域のサイズを設定します。

Hf(128)[byte] ファイル名ヒストリーバッファの空き領域のサイズを設定します。

Hx(128)[byte] DOSコマンドラインヒストリーバッファの空き領域のサイズを設定します。

Ha(64) [byte] アプリケーションコマンドヒストリーバッファの空き領域のサイズを設定します。これは、symdeb等のアプリケーションでコマンドを入力する際に使用されます。

Bl(1024)[byte] 行バッファのサイズを設定します。
[範囲] 1024 ～ Bt*1024/8

Bc(256)[byte] 削除文字列バッファのサイズを設定します。
[範囲] 256 ～

Bu(64) [byte] マクロ用配列・スタック領域のサイズを設定します。
[範囲] 32 ～

■ 各種モード

GV(0)　　　裏グラフィックVRAM使用モードを設定します。
　　GV0　:使用せず
　　GV1　:A800h ～ B7FFh
　　GV2　:B000h ～ BFFFh
　　GV3　:B800h ～ BFFFh , E000h ～ E7FFh
32+**Bo**[KB]のワークをVRAM上に割り当てます。内訳は
　　・スクリーンの待避エリア
　　・ファイラーのディレクトリバッファ（オプションFW)
　　・コンソール出力バッファ（オプション**Bo**)
です。このサイズ分だけ、メモリの消費が少なくてすみます。
※ PC-9801ノーマルモード以外では使用できません。
※　裏VRAMを持たない機種では、表VRAMを使用します。
※「**GV**3」は、16色表示の可能な機種のみ使用できます。

EM　　1.5　EMSの使用ページ数を設定します。
　　EM0　:EMSメモリは使用しない
　　EMn　:EMSメモリを n ページ使用する
　　EM　:EMSメモリの残りを全て使用する
※　いずれも最大64ページ（1MB）までです。

SW1　* 1.5　スワップモードを設定します。
　　SW0　:スワップ機能を使用しない
　　SW1　:テキスト領域をスワップアウトする
　　SW2　:コード領域までスワップアウトする
　　SW3　:常駐直後にコード領域までスワップアウトする

TC(10)　　オープン可能なテキスト数を設定します。
　　[範囲] 1 ～
　　[メモリ消費] 210* **TC** byte

WD(80) * 1.5　デフォルトの表示カラム数を設定します。

WE(80) * 1.5　現在編集中のテキストの表示カラム数を設定します。
　　[範囲] 2 ～ 250

PG(0)　*　　表示行番号モードでの1ページの行数を設定します。
　　PG-　:ページ表示せず

RS(3)　*　　スムーススクロールの速度を設定します。
　　[範囲]（低速）0 ～ 4（高速）

Ta(8)　*　　タブカラム数を設定します。

Tb(4)　*　　.c .h ファイルのタブカラム数を設定します。

[範囲] 1, 2, 4, 8, 16, 32, 64

Ci(4) ＊ 挿入モード（Insert）時のカーソルの形状を設定します。

Co(7) ＊ 上書モード（Overwrite）時のカーソルの形状を設定します。

+8	+4	+2	+1	
x	x	0	0	フルサイズ
x	x	0	1	1/2 サイズ
x	x	1	0	1/4 サイズ
x	x	1	1	アンダーバー
0	0	x	x	常時点灯
0	1	x	x	通常速度の点滅
1	0	x	x	1.5倍速の点滅 ※
1	1	x	x	2倍速の点滅 ※

※ PC-9801版のみ

FW(256) ファイラーで扱うことできるディレクトリの最大数を設定します。

[メモリ消費] 22＊**FW** byte

FV(1) ＊ ファイラーのディレクトリ表示モードを設定します。

FV1 :ロング（縦分割）

FV5 :ショート（横分割）

FS(0) ＊ 1.5 ファイラーSortメニューのオプションの初期値を設定します。

FS1 :ファイル名

FS2 :拡張子

FS4 :新しい順

FS8 :古い順

FS16 :大きい順

FS32 :小さい順

FS64 :ディレクトリを先頭へ

FO(3) ＊ ファイラーCompareメニューのオプションの初期値を設定します。

+16	+8	+4	+2	+1	
x	x	x	x	1	存在
x	x	x	1	x	新しいファイル
x	x	1	x	x	古いファイル
x	1	x	x	x	大きいファイル
1	x	x	x	x	小さいファイル

WL(0) ＊ ポップアップウィンドウの表示位置を設定します。

+8	+4	+2	+1	
x	x	x	1	←よせ

x x 1 x　↑よせ
x 1 x x　Hセンタリング
1 x x x　Vセンタリング
設定しない場合は、カーソル位置を基準とします。

LC(-)　編集モード時の画面行数を指定します。
何も指定しない時(-)は、起動時の行数になります

■ 表示属性

An(7)	*	通常のテキスト
Ac(5)	*	テキストのCR, TAB, EOF文字
Al(5)	*	行番号
Ah(13)	*	ブロック領域
As(15)	*	ステータスライン
Ao(14)	*	コンソールファイルのステータスライ
Ar(13)	*	Read onlyファイルのステータスライン
Am(6)	*	最下行のメッセージ
Ab(5)	*	ウィンドウわく
At(5)	*	ウィンドウタイトル
Aw(7)	*	ウィンドウメニュー
Af(6)	*	ウィンドウメニューのダイレクトキー
Ai(15)	*	ウィンドウメニューのカーソル
Aj(6)	*	ヘルプウィンドウ
Ak(6)	*	2ストローク先頭キー
Ag(1)	*	ファイラーの背景
Au(6)	*	ファイラーのカーソル
Ap($70)	* 1.5	ファンクションキー（J-3100,IBM PC版のみ）

<PC-9801版>
0:黒 1:青 2:赤 3:紫 4:緑 5:水色 6:黄 7:白
+8:リバース +16:アンダーライン

<J-3100版>
6:ボールド 7:ノーマル +$70:リバース +8:アンダーライン

<IBM-PCカラー>
0:黒 1:青 2:緑 3:水色 4:赤 5:紫 6:黄 7:白 （低輝度）
8:黒 9:青 $A:緑 $B:水色 $C:赤 $D:紫 $E:黄 $F:白 （高輝度）
背景色はx16

<IBM-PCモノクロ>
0:黒 7:白 +8:高輝度／ボールド +$70:背景白

■ スイッチ

Dl(-) ＊ 行番号を表示します。
『L 行番号の表示』

Dc(+) ＊ CR 文字を表示します。
『C ↵ の表示』

Dt(-) ＊ TAB文字を表示します。
『T TABの表示』

De(+) ＊ EOF文字を表示します。
『E EOFの表示』

Ds(-) ＊ ステータスラインへシステム情報を表示します。
『S システム情報の表示』

Db(-) ＊ ¥文字をバックスラッシュ（＼）で表示します。
『¥ バックスラッシュ』
※ フォントをもっていない機種では使用できません。

Dn(-) ＊ 行番号モードを切り替えます。
Dn- :論理行番号
改行コード（0Ah）から次の改行コードまでを1行とした時の行番号。
Dn+ :表示行番号
画面上に表示される1行をそのまま1行とした時の行番号
『N 行番号モード』

Du(-) ＊ カーソル行アンダーラインを表示します。
『U アンダーライン』

Df(-) ＊ 編集ファイル名を、フルパス名で表示します。

Dp(-) ＊ 全てのファイル名・パス名を小文字で表示します。

Ei(+) ＊ オートインデントを指定します。
『A オートインデント』

Es(+) ＊ ^QE／^QX コマンドを、それぞれ画面上端／下端で実行した時にスムーススクロールを起動します。

Ez(+) ＊ 書き込むファイルの最後に、EOFコード（1Ah）を付加します。

Eb(+) ＊ ファイルの書き込み時に、バックアップファイルを作成します。
『B .BAK ファイルの作成』

Eu(-)	*	[TAB]をスペースに変換して入力します。
Et(+)	*	論理ページング機能（Cモード）を使用します。
Eh(-)	*	ページング（Pモード）の画面単位を切り替えます。 ***Eh***⁻　:全画面 ***Eh***⁺　:半画面 『H ページング単位』
Ew(+)	*	文字列検索時のデフォルトをワードサーチにします。 『W ワードサーチ』
Ec(-)	*	文字列検索時に英大文字と小文字の違いを無視します。 『I 検索時の英大/小文字』
Ex(-)	*	テキストスタックが一杯になった時、警告します。
En(-)	*	カーソルを行末以降に表示させなくします。
Ev(-)	*	ジャンプ後、カーソルを画面の中央へ移動します。
Ek(-)	*	^J ^KCコマンドでブロック挿入後、カーソルをブロックエンドに移動します。
Ea(-)	* 1.5	オープンしたファイルの拡張子が数字の場合、表示カラム数をその値に設定します。
RO(-)	* 1.5	Viewモードでオープンします。
Sx(+)	*	DOSコンソール出力の取込みを行ないます。
Se(+)	*	DOSコンソール出力の取込み時に、エスケープシーケンスをスキップします。
Sd(+)	*	DOSコマンドを実行前にエイリアスで展開して再表示します。
Sk(-)	*	DOSコマンド実行前に、テキストスタックをクリアします。
Sr(-)	*	[ESC]Q 終了時、「editfile」に編集情報をセーブします。
Sy(-)	*	(Y/N) 入力時、[↵]=Y, [SPACE]=N で 答えられるようにします。
St(-)		DEFファイルを読み込み時に、先頭の1行を画面に表示します。
Sn(+)	*	編集ファイルが見つからない場合の確認動作を指定します。 ***Sn***⁻　:ただちに新規ファイルとしてオープン ***Sn***⁺　:新規ファイルかどうか確認
Sq(-)	*	[ESC]C, [ESC]Qコマンドで即座にエディタを終了します。
FP(-)		FEP（VJE β、松茸、ATOK）を制御します。

FR(-)　1.5　起動後ただちにFEPをONします。(起動モードを引数で指定)

FT(0)　FEPの自動識別が使えない場合に、FEPのタイプを指定します。

<PC-9801版>

FT0　:自動識別

FT1　:VJE β

FT2　:松茸

FT3　:ATOK

<J-3100版>

FT0　:自動識別

FT1　:ATOK7

FT2　:ATOK5/6

FK(-)　1.5　ファンクションキーのモードを指定します。

<PC-9801版>

FK+　:[**f･11**]〜[**f･15**]を使用する

(ノーマル/ハイレゾ共用DOSのみ動作)

<J-3100,IBM PC版>

FK+　:12ファンクションキー表示

第11章　マクロ機能

11.1 マクロの概要
11.2 キーボードマクロ
11.3 外部マクロの実行
11.4 マクロのコーディング
11.5 マクロ言語仕様

11.1 マクロの概要

ひとくちに「マクロ」といっても、アプリケーションの種類によって様々な意味をもちます。エディタの場合は、ふつう次のどちらかの機能を指します。

・一連のキー操作を1つのキーに記憶し、ワンタッチで実行させる機能
　　→　キーボードマクロ
・マクロ言語を用いて、コマンドをプログラミングできる機能
　　→　プログラマブルマクロ

VZではどちらの機能もサポートしています。特にマクロ言語はたいへん強力です。編集機能のうち、基本コマンドはシンプルで汎用性のあるものに限定し、複合的な機能はすべてマクロで実現する、というのがVZの基本方針です。このため、マクロで組んだ機能は、基本コマンドと全く同等に扱うことができます。

■ VZのマクロ言語

いくら強力な言語仕様であっても、それが難解では意味がありません。マクロ言語にはLisp風のものも幾つかありますが、これはLispに親しんだ者でなければ、容易に理解できません。VZのマクロ言語は、一見すると記号が多くて難解に思われますが、これはキーワードを「記号＋英字1字」に短縮して表現しているためです。しかし、演算式はC言語に準じた馴染みやすいものですし、また変数名も「英字2字」で簡潔に表現していますので、C言語をある程度使える方は、すぐにプログラムできるでしょう。

■ マクロのコーディング環境

VZのマクロは、コンパイル等の面倒な作業を一切必要としません。DEFファイルの「* M マクロ」の項にマクロコードをそのまま記述します。従って、マクロをコーディングしつつ、再カスタマイズ（^K^）で組み込み、即座に実行してみることができますから、まさにBASICインタプリタの感覚でデバッグできます。

■ キーボードマクロ

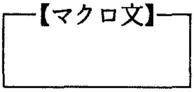
キーボードマクロで登録したキー操作も、内部でマクロコードに変換して格納するため、登録後は他のマクロと同様に使用できます。また、登録したマクロコードは、編集テキスト中に挿入するコマンド（^K_）を利用して、これをDEFファイル中に挿入し、次回からはマクロコマンドとして使用することもできます。

■　インスタントマクロ

マクロコードは、ほとんど記述したままの形式でメモリに格納します。従って、キーボードから入力した文字列をマクロコードとみなして実行することも可能です。

[ESC][@]　　　　　　　:マクロ文の実行（マクロ）

入力した文字列をマクロコードとみなして実行し、返り値を10進・16進・文字・文字列で表示します。コマンドを実行すると、



というウィンドウが開きますので、次の例のような演算式を入力してみてください。

[例]

　　1234+$5678　'漢'　an=ac=6　du+　ef

オプションは変数と同様に扱えますから、このコマンドで値を参照したり変更することができます。

11.2 キーボードマクロ

マクロのプログラミングなんて難しいから必要ない、という方も、「キーボードマクロ機能」だけは覚えておいて損はありません。

キーボードマクロとは、一連のキー操作を1つのキーに記憶し、ワンタッチで実行させる機能のことです。

例えば、「list. txt」を「list. bat」に変換したいとしましょう。

list. txt

```
VZ      COM   40200  89-06-30   1:01
VZ      DEF   11178  89-06-30   1:01
VZ      DOC   64445  89-05-10   1:00
```

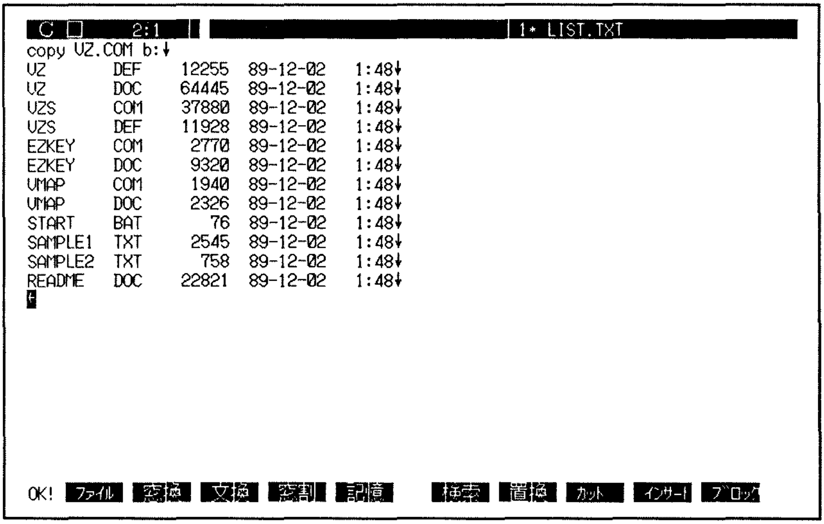
list. bat

```
copy VZ.COM b:
copy VZ.DEF b:
copy VZ.DOC b:
```

このように、同じ手順の変換を各行で行なう場合に、キーボードマクロが役立ちます。
まず、最初の行の変換を次の手順で登録します。

- 「list. txt」の先頭にカーソルを移動
- [CTRL]+[_]でキーボードマクロの登録を開始
  (この時画面の左下に「m」が表示されます)
- "copy " ^F [SHIFT]+[BS] "." ^F [SHFIT]+[BS] ^QY " b:"
- ^QS [↓]でカーソルを次の行の先頭へ移動
- [CTRL]+[¥]でキーボードマクロを登録
  (この時画面の左下に「OK!」が表示されます)

一連のキー操作を、[CTRL]+[¥]キーに登録したわけです。そうしたら、[CTRL]+[¥]を押してください。2行目が瞬時に変換されました。後は[CTRL]+[¥]を押し続ければ、いくらでも変換できます。



```
 C □    2:1                              1* LIST.TXT
copy UZ.COM b:↓
UZ        DEF    12255  89-12-02   1:48↓
UZ        DOC    64445  89-12-02   1:48↓
UZS       COM    37880  89-12-02   1:48↓
UZS       DEF    11928  89-12-02   1:48↓
EZKEY     COM     2770  89-12-02   1:48↓
EZKEY     DOC     9320  89-12-02   1:48↓
UMAP      COM     1940  89-12-02   1:48↓
UMAP      DOC     2326  89-12-02   1:48↓
START     BAT       76  89-12-02   1:48↓
SAMPLE1   TXT     2545  89-12-02   1:48↓
SAMPLE2   TXT      758  89-12-02   1:48↓
README    DOC    22821  89-12-02   1:48↓

OK! ﾌｧｲﾙ 窓換 文換 窓割 記憶  検索 置換 ｶｯﾄ ｲﾝｻｰﾄ ﾌﾞﾛｯｸ
```

[CTRL]+[_]　　　　　:キーボードマクロ

キーボードマクロの登録を開始します。コマンドを実行すると、ファンクションキー表示の左端に、「m」を表示して、キーボードマクロの登録中であることを示します。この後、以下のキーが入力されるまでの一連のキー操作を登録します。

[CTRL]+[_]　　　　　　　：登録を中止
[CTRL]+[¥]等の未定義キー：登録

登録が完了すると、「OK!」と表示し、これ以降[CTRL]+[¥]キーで一連の操作が実行できます。異なった未定義キーを使えば、複数のキーボードマクロを登録することもできます。また、同じキーに何回でも登録できます。

キーボードマクロは、「オプション***Bm***」で指定したマクロバッファのフリーエリアに登録します。登録中にバッファが一杯になった場合は、自動的に登録は中止し、「m」は消されて、それまでの登録は無効になります。

キーボードマクロの登録中に他のマクロを実行しても構いません。

登録したキーボードマクロをセーブしておきたい場合もあります。

[CTRL]+[K][_]　　　:キーボードマクロ文挿入

　このコマンドを実行すると、現在までに登録した全てのキーボードマクロを、DEFファイルに記述するマクロコードに直してカーソル位置に挿入します。
先の例で登録した場合は、次のような文字列になります。

```
90 ^¥ "" "copy "#f#16"."#f#16#1" b:"#<#x
```

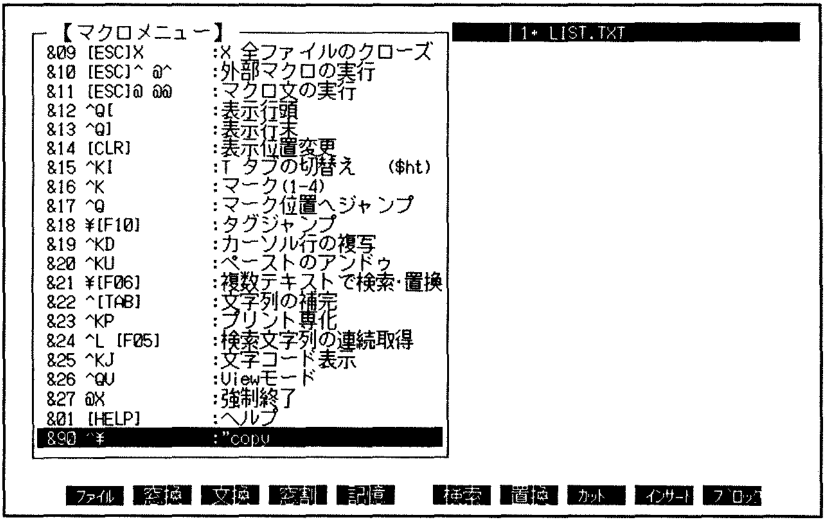
これをDEFファイルの「* M マクロ」の適当なところへコピーすれば、次回からマクロコマンドとして使うことができます。もちろんその場合は、番号とキー定義を適当なものに変更する必要があります。

　キーボードマクロには、90番～99番のマクロ番号を割り当てています。すなわち、最大10種類のキーボードマクロを同時に登録できることになります。逆にマクロ番号は89番までしか使えません。

現在、定義されているマクロの一覧を表示することもできます。

[**ESC**] **M**　　　　　　　　　　:マクロメニュー

現在設定されている全マクロのメニューを表示します。[↵]で選択すると、そのマクロを実行します。



11.3 外部マクロの実行

拡張子が「.def」のファイルのうちで、以下のファイルは「マクロファイル」と呼ばれるマクロコマンドを定義したファイルです。

tool.def　block.def　keisen.def　zenhan.def　game.def ...

これらは、VZの拡張機能として、必要に応じて組み込んで使用します。例えば、起動時に

A>vz +block

のように指定できます。さらに、次のコマンドで編集中に読み込むこともできます。

[**ESC**][^]　　　　　:外部マクロの実行（マクロ）

コマンドを実行すると、ファイラーが起動し、DEFファイルのディレクトリを表示します。ここで、組み込みたいマクロファイルを選択して[CR]を押すと、そのファイルを読み込み、最初のマクロを実行します。

```
 Drive Mask Path Sort View Window All TrCD RetCD Compare Exec
┌ A:¥*.def ─────────────────────────────────┐<
│ J31¥               <DIR>  89-12-02  1:48 │イトル
│ IBM¥               <DIR>  89-12-02  1:48 │字
│ SRC¥               <DIR>  89-12-02  1:48 │頭文字
│ VZ       .DEF      12255  89-12-02  1:48 │ーソル
│ VZS      .DEF      11928  89-12-02  1:48 │ドウ
│ VZE      .DEF       9254  89-12-02  1:48 │
│ UPDATE   .DEF        424  89-12-02  1:48 │景
│ HELP     .DEF       1709  89-12-02  1:48 │ーソル
│ HELPE    .DEF       1587  89-12-02  1:48 │
│ WS       .DEF      10019  89-12-02  1:48 │
│ MI       .DEF      14682  89-12-02  1:48 │
│ CMOS     .DEF      10535  89-12-02  1:48 │
│ TOOL     .DEF        975  89-12-02  1:48 │
│ BLOCK    .DEF       2287  89-12-02  1:48 │表示
│ KEISEN   .DEF       3762  89-12-02  1:48 │
│ ZENHAN   .DEF       1023  89-12-02  1:48 │号
│ GAME     .DEF       2573  89-12-02  1:48 │ダーライン
│ ---------- <End of Dir> ----------       │示
└── 17 files ─ 412672 bytes free ──────────┘文字表示
↓
Ei+                         ;オートインデント↓
Es+                         ;^QE,^QX でスムーススクロール↓
   C1   CU   CA   S1   SU      VOID  NWL  INS  REP  ^Z
```

このコマンド自体は、『10.4 編集時の再カスタマイズ』でふれた

[CTRL]+[K][^]　　　:再カスタマイズ

を応用しているにすぎません。従って、マクロのフリーエリアが足りない場合には、

《バッファがいっぱいです》

と表示し、正常に組み込めませんので、「オプション ***Ba***」で十分なバッファを確保しておかなければなりません。また、マクロを組み込んだ後自動的に実行するマクロは、ファイルの中で最初に現れたマクロです。

●●　マクロファイルの分割　●●

マクロの数が増えて、これらを全てDEFファイルに記述していると、コード領域（最大64KB）がいっぱいになり、起動時に

メモリが足りません.

と表示され、VZが立ち上がらなくなることがあります。特にVer1.5になり、コードサイズが5KBほど大きくなりましたので、このために立ち上がらなくなる場合も考えられます。このような場合には、マクロの部分をいくつかのファイルに分けてください。あまり使わないマクロをマクロファイルに分け、必要な時に「外部マクロの実行機能」で読み込んで使用すれば、メモリサイズも小さくなり、同時に使えるマクロも多くなります。

このような場合には、併用して用いることのないマクロファイルは、それぞれ最初のマクロ番号を同じにしておくとよいでしょう。こうすると、次々にマクロを読み込む際に、以前読み込んだマクロ領域の上にオーバーライトしますから、マクロバッファを効率的に使うことができます。

DEF ファイル

```
┌ VZ.DEF ─────────┐
│* M マクロ        │
│...               │
│40 最後のマクロ   │
└──────────────────┘
```

マクロファイル

```
┌ MAC1.DEF ───────┐   ┌ MAC2.DEF ───────┐
│* M マクロ        │   │* M マクロ        │
│50 "外部マクロ1   │   │50 "外部マクロ2   │   ...
│...               │   │...               │
└──────────────────┘   └──────────────────┘
```

11.4 マクロのコーディング

VZはかなり高度な編集機能をもっていますが、だからといって全てのユーザーのニーズに応えることはできません。エディタの使い方は、それがベーシックなツールであるがゆえに本当に様々です。そして、これを少しでも補うべく考え出されたのが「マクロ言語」です。

VZのマクロ言語は、コンパクトですがたいへん柔軟で、かつ強力なものです。C言語をご存じの方はすんなりと馴染めることと思います。それでは、簡単な例をもとに、マクロコードのコーディングとデバッグの方法について説明していきましょう。

■　どんなマクロを作るか

簡単でしかもある程度実用になる例として、

「指定コード範囲の文字の検索」

を作ってみましょう。これは、「0~9」までとか、「A~Z」まで、というように文字範囲を指定すると、該当する文字をカーソル位置から検索する、というマクロです。検索文字列に正規表現が記述できれば、このような検索はコマンドでできるのですが、残念ながらVZではサポートしていません。

■　マクロファイルを作る

何はともあれ、マクロを書くためのファイルをオープンします。これは新規ファイルですから、[**ESC**]**N** で開きましょう。ファイル名は、「search.def」とでもしてください。

新規ファイルをオープンしたら、まず次のようなリストを打ち込みます。

```
1: ===== search.def =====
2: * M マクロ
3: 80 ^¥ "Search"
4:
5: *
6: ===== End of search.def =====
```

便宜上行番号を付けておきます。これはマクロファイルの一般的な書式です。

1: 最初の行には必ずコメントを書きます。

2: 「*」の後の「M」で、マクロの開始を表します。

3: マクロの通し番号、および実行するキー（未定義のもの）を指定します。
4: この間にマクロコードを書きます。
5: 「*」のみの行は定義の終了を意味し、これ以降は読みません。
6: テキストの最後を示すコメントです。「*」の後ろなので無視されます。

4行目からマクロコードを書き始めます。マクロコードの各キーワードは、スペース、タブ、改行で区切ることができます。

■　どんなコマンド・関数を使うか

仕様が決まったら、それをどのようなVZのコマンド、またはシステム関数で実現するかを検討します。このためには、どのような機能があるのかひと通り知っておかねばなりません。『11.5 マクロ言語仕様』にざっと目を通してください。一般的なプログラミングも、実はこれが最も重要なポイントなのです。このマクロでは、次のような機能が必要になると予想されます。

(A) キーボードからの文字コードの入力
(B) 文字コードの変数への格納
(C) カーソル位置の文字コードの取得と判定
(D) カーソル位置の更新と、テキストの終端の判定

この例では、「指定範囲の文字が見つかるまでカーソルを進める」という、至極単純な方法を用いています。

■　文字コードの入力(A)

キーボードから文字列を入力する場合は、次の関数を使います。

【 &g("...")　:文字列の入力 】

これは、指定タイトルの文字列入力ウィンドウを開き、入力した文字列、終了キーに応じて値を返す、というものです。文字列は、「ポインタ変数pw」が示すワークバッファに格納します。ちょっと実際に書いて試してみましょう。

```
1: ===== search.def =====
2: * M マクロ
3: 80 ^¥ "Search"
4:     &g("文字検索(A-Z):")
5:     &m("r=%d, pw=$"%s$"",r,pw)
6: *
```

```
7: ===== End of search.def =====
```

5行目の「&m("...」は、C言語のprintf関数に準じたシステム関数で、最下行に所定のメッセージとして表示します。

■　マクロの組み込み

実は、この例もすでにマクロファイルになっています。[**CTRL**]+[**K**][^]を押しましょう。画面最下行に

《1 項目読み込みました》

と表示されましたね。たったこれだけでマクロの組み込みは完了です。それでは[**CTRL**]+[¥]を実行してみます。画面に「&g」で設定したウィンドウが開きます。

```
┌ 文字検索(A-Z): ┐
│              │
└──────────────┘
```

「abc」とか入力して[↵]を押すと、画面最下行に、それぞれのレジスタ（r,pw）の内容を表示します。

《r=0, pw="abc"》

いろいろな文字列を入力して、どんな値が返されるか試してください。こんな具合に、リアルタイムでそれぞれの関数の機能を調べることができます。

■　文字列の解析(B)

さて、「&g」で入力した文字列を解析します。ここでは、文字列を3文字の「文字1」,「区切り文字」,「文字2」として入力することにします。文字列は、「ポインタ変数pw」の示すワークに格納されています。よってポインタ演算子を使い、

```
a=pw.0,
b=pw.2,
```

とすることで、「文字1」,「文字2」を、それぞれ「変数a,b」にロードできます。C言語では「;」で式を区切りますが、VZのマクロでは「;」はコメントに用いますので、代わりに「,」で区切ります。また、条件式などで式の値を評価する場合は、(...)で括ってください。

ところで、この方法では、文字として漢字を指定できません。次のように漢字かどうか判定して、漢字ならばワードポインタとして読み込めばよいのですが、少々難しくなりますので、ここでは半角のみに限定します。また、不正な文字を入力した時のエラーチェックも省きます。

```
a=pw.0, (a>$80 && a<$a0 || a>$DF && a<$FD)?{ a=pw..0, }
```

■ 文字コードの取得と判定(C)

範囲を決める変数が取得できましたので、今度はカーソル位置の文字コードの取得です。これは「システム変数***cd***」の値をみるだけです。

【 ***cd*** :カーソル位置の文字コード 】

VZのマクロでは、各種情報はオプションやシステム変数を直接参照するだけで得られます。従って、カーソル位置の文字が「変数a」と「変数b」の間の文字かどうかは、

```
(cd>=a && cd <=b)?{ ... }
```

という条件式で判定できます。この演算式は、(...)の中が「真」の時、{ ... }を実行するもので、条件式の基本です。慣れないうちは、

```
(cd>=a && cd <=b)? >A
```

という具合に、「if...goto...」式に記述するのが、見栄えは悪くても安全で確実な方法です。「>A」は、ラベル「:A」へジャンプするジャンプ命令です。

■ カーソルの更新(D)

カーソル位置の文字が指定範囲内の文字でなければ、カーソルを進めます。カーソルを進めるのは、**^D** キーに相当する「#d」コマンドです。このように、基本コマンドの多くは、標準キーから連想できる「英字1文字」で表せます。もしコマンドが判らなくなったら、[**HELP**]キーを利用してください。[**HELP**]キー機能そのものもマクロで記述してありますが、次のような便利な機能も持っています。

- 押したキーの1ラインヘルプを表示する。
- 「#」の上で[**HELP**]を押すと、続くコマンドの1ラインヘルプを表示する。

さて、カーソルを進める場合、テキストの最後に達したかどうかチェックしなければなりません。これには二とおりの方法があります。

- カーソル位置の文字タイプ「変数ct」がEOF (0) か？
- 「#d」で進めた時、コマンド実行後の返り値「r」が (0) でないか？

ここでは後者の方法を使います。

■ ファーストバージョン

では、以上をまとめて1つのマクロコードを組んでみましょう。

```
===== search.def =====
* M マクロ
80 ^¥ "Search"
     &g("文字検索(A-Z):") (r<0)?.
     a=pw.0, b=pw.2,
:A   #d (r<0)? .
     (cd<a || cd>b)? >A
     &m("Found!")
*
===== End of search.def =====
```

組み終わったら、[CTRL]+[K][^]で組み込み、[CTRL]+[¥]で実行します。適当に文字の範囲を指定すると、カーソル位置の前方にサーチしていき、見つかると

《Found!》

と表示します。

■ いくつかの改良

さて、使っていくうちにいくつか気になる点が出てくると思います。

(1) 遅い
(2) 毎回検索文字を指定しなくてはならない
(3) 見つからないとテキストの最後に行ってしまう
(4) 途中で中断できない。

まずスピードですが、全てのマクロで使える手段として、

「処理中の画面表示を停止する」

方法があります。通常は、1コマンド実行する度にステータスラインを更新していますので、これを表示しない分だけ速くなります。これは、次の関数で制御します。

【 &s　:画面表示の停止 】
【 &d　:画面表示を再開し全画面を再表示 】
【 #?　:画面表示を再開しカーソル行を再表示 】

(2)は、[↵]のみ押した場合に、以前の入力文字列を使うことで対応します。[↵]のみ入力すると「返り値r=-2」となります。
(3)は、検索の前にカーソル位置をマークしておくことで対応します。
(4)は、とりあえず現状のままでも次のキーで中断できます。

[**STOP**]	:マクロの中断
[**Ctrl**]+[**Break**]	(J-3100, IBM PC)

マクロの実行を中断し、トレースモードに入ります。ファンクションキー表示の左端に、中断した位置のマクロコードの一部を表示し、次のキーの入力待ちになります。

[**SPACE**] ･････････ ステップ実行
[↵] ････････････ マクロの連続実行を再開
[**ESC**] ･･･････････ マクロ実行の中止

しかし、ループの中でキー入力チェックを入れた方がよいでしょう。

【 &k　: BIOSキー入力チェック 】

を使って、キーが押されたがどうかをチェックします。

■　セカンドバージョン

以上の改良を加えた新しいマクロは、次のようになります。

```
===== search.def =====

* M マクロ

80 ^¥ "Search"
      &g("文字検索(A-Z):")              ; 検索文字を入力
      (r==-1)?.
      (r!=-2)?{ a=pw.0, b=pw.2, }       ; [↵]のみの場合は設定しない
      #M                                ; カーソル位置をマーク
```

```
        &m("Search %s",pw)          ; 検索文字コードを表示
        &s                          ; 画面表示を停止
   :A   &k (r!=0)? >B               ; キーが押されたら中断
        #d (r<0)? >N                ; テキストの最後に達したら終了
        (cd<a || cd>b)? >A          ; カーソル位置の文字と比較
        &d &m("Found!") .
   :N   &d &m("Not found.") .
   :B   &d &m("Break!") .

 *
 ===== End of search.def =====
```

■　ブレークポイント

マクロをデバッグする際に、指定箇所で中断して様子を見たい場合があります。このような場合は、「&t」を中断したい箇所に挿入してください。

【 &t　　:トレースモード 】

「&t」のところへ達すると、マクロの実行を中断し、トレースモードに入ります。この後の操作は、[**STOP**]で中断した場合と同様です。

以上のように、VZのマクロは書いたその場で組み込み・デバッグができますので、開発はたいへん容易です。どうぞ実際にマクロを作ってお試しください。

11.5 マクロ言語仕様

VZのマクロ言語は、次のような特徴を備えています。

- 39種のCライクな演算子（push, pop, byte/word ptr等のオリジナル演算子を含む）
- 56個の汎用変数の他、オプション変数、システム変数を演算式で使用可能
- 24種のシステム関数（printf, int86, strchr, locate, rand相当の関数を含む）
- DEFファイルで定義したポップアップメニューを呼び出し可能
- if..else.. on..goto.. に相当する条件分岐、ブロック、ループ制御構造
- 最大8レベルのネストおよびバックグラウンドでのマクロ実行が可能

また、マクロコードは、テキストの状態のままでメモリに格納するため、コマンド・変数・ラベル名等は1、2文字の英数字で表しています。その結果、多少読みづらい点がありますが、それを補って余りある次のような特徴を有します。

- 編集中のマクロコードを即座にマクロバッファへ読み込み、実行できます。
  （^K^ コマンド）
- トレース時に、実行中のマクロコードの一部を、画面に表示します。
- 任意の文字列をマクロコードとしてコールすることができます。
  （[ESC]@ コマンドは、この機能を利用しています。）

キーワード

VZのマクロコードは、次にあげる「キーワード」と、後で述べる「演算式」で構成されています。各々のキーワードの間は、スペース・タブ・改行で区切っても、つなげて記述しても構いません。ただし、一つのキーワード中には、このような区切り文字を挿入してはなりません。

以下の書式では、

n	1文字の10進数
a	1文字の英小文字
A	1文字の英大文字

を表します。

"<string>"

<string>をキーボードからの入力文字列とします。

'<char>'

<char>の1文字（半角／全角）をキーボードからの入力文字とします。

#nn　#a　#A

nn番の基本コマンドを実行します。多くのコマンドは、番号の代りに、より覚えやすい1文字の英文字または英記号で記述できます。

変数 r には、コマンドの返り値をセットします。通常の返り値は次のとおりです。

r == 0 ・・・・・・ 正常終了
r == -1 ・・・・・・ 実行不可能／キャンセル／エラー等

詳しくは、「付録A.基本コマンド一覧」を参照してください。

!nn

nn番のポップアップメニューを呼び出します。

変数 r には、次の値をセットします。

r >= 0 ・・・・・・ 選択したメニューアイテムの番号
r == -1 ・・・・・・ キャンセル

nn番のメニューをサーチする場合、まず現在実行中のマクロ以降のバッファで探し、見つからない場合には、バッファの先頭に戻って探します。システムメニューは、呼び出すことができません。

&nn &>　&*　&+n &−n

指定マクロをコールします。(8レベルまでネスト可能)
変数 r には、呼び出したマクロで最後に（式）で評価した値をセットします。
各々の書式の指定するマクロは、次のとおりです。

&nn	nn番のマクロ
&>	次のマクロ
&*	最後に組込んだDEFファイルで最初に現れたマクロ
&+n &-n	現在実行中のマクロ番号±n番のマクロ

nn番のマクロをサーチする場合、まず現在実行中のマクロ以降のバッファ内の、ローカルマクロを探し、見つからない場合には、バッファの先頭からグローバルマクロを探します。

&a &a(...)

システム関数をコールします。関数によっては、変数 r に返り値をセットします。
各々のシステム関数は、この後で説明します。

&#nn(...) &#a(...) &#A(...)

基本コマンドの一部は、このように引数を持ってマクロから呼ぶことができます。これらのコマンドは、「コマンド関数」と呼びます。
各々のコマンド関数は、この後で説明します。

>nn >>　>*　>+n >−n

指定マクロへジャンプします。各々の書式の内容、およびサーチの手順は、コールの場合と同様です。

>^

実行中のマクロの先頭へジャンプします。

>A

指定ラベルへジャンプします。

:A

ラベルです。ラベルは、1文字の英大文字で記述します。

.

実行中のマクロを終了します。サブルーチンからの return に相当します。

／

マクロの実行を中止します。

(式)

(...)内の演算式を評価し、結果を内部レジスタに格納します。

式,

キーワードが英字で始まった場合、それ以降、文末または ,(カンマ)までを演算式として処理します。(式)の場合と異なり、式の値の評価は行ないません。

?

条件分岐キーワードです。内部レジスタの値が0なら、続く1キーワードまたはブロックをスキップします。その後内部レジスタの値を論理NOTします。

{

ブロック開始キーワードです。内部レジスタの値をループカウンタにセットし、カレントポインタをループ開始ポインタにセットします。

}

ブロック終了キーワードです。次のように処理します。(count:ループカウンタ)

```
if ( count ) {
    if( count == -1 || --count ) {
        <ループ開始ポインタへジャンプ>
    }
}
```

>?

多重分岐キーワードです。内部レジスタの値がnの時、続くブロック内でn番目のキーワードまたはブロックを実行します。実行後、および該当する項目がない場合は、ブロックの次のキーワードから実行します。

これらの制御キーワードを用いて、次のような制御構造が記述できます。

(式)　? <true>　? <false>

条件文です。(式)の値が真（0でない）なら <true> を、偽（0）なら <false>を実行します。<true>, <false>はキーワードまたはブロックです。

★注意★

<true>の内部で(式)を記述すると内部レジスタを再設定してしまい、続く<false>が正常にスキップしません。同様の理由で、条件分岐をネストすることもできません。

(式) {...}

ループ文です。(式)の値回数、ブロックを実行します。式の値が0の場合でも、1回だけ実行します。(-1){...}は無限ループとなります。

★注意★

ループブロックの中に、ブロックをネストさせてはなりません。

(式)　>? { <if0> <if1> <if2> ...　}

多重分岐文です。(式)の値が n の場合、<ifn>を実行します。<ifn>は、キーワードまたはブロックです。

演算式

演算式は、定数、変数、演算子で構成されます。

■ 定数

定数として、10進数、16進数、文字定数、文字列、コマンドコードが使用できます。

```
10進数 10   32767  -1
16進数 $0D  $8140
文字定数'A'  'あ'
文字列        "string"  "文字列"
コマンドコード  #70 #m
```

・定数は、int（2バイト）で表します。

・文字列は、終端の " を0に変換して、バッファに格納します。また、文字列の値は、先頭文字のオフセットです。

・コマンドコードは、コマンド番号に256を加えた値です。

■ 変数

変数は、1、2文字の英字（大／小文字は区別しない）で表し、汎用変数、汎用フラグ、オプション変数、システム変数に分類されます。

■ 汎用変数

汎用変数として、次のものが使用できます。

・r, s, w を除く a ～ y

・rr, ss, ww を除く aa ～ yy

・ax, bx, cx, dx, si, di （割り込み処理関数での受け渡し変数として使用）

いずれも int型（2バイト）です。起動時にはすべて0で初期化しています。マクロ実行時には初期化することはありません。

■ 汎用フラグ

汎用フラグとして、fa, fb, fc, fd, fx, fy が使用できます。汎用フラグは、オプションのスイッチと同様に、0（偽）または1（真）の値をとります。起動時はすべて0（偽）です。

■　オプション変数

起動時オプションも、演算式で他の変数と同様に使用できます。ただし、実行中に変更可能なオプション以外のものは、値を変更してはなりません。

■　システム変数

エディタ内部の種々の状態を、システム変数で参照、変更することができます。各々のシステム変数は、この後で説明します。

■　演算子

演算子は、単項演算子、二項演算子、代入演算子に分類できます。その多くはC言語の演算子に準じますが、いくつは独自の仕様です。これらの演算子は、以下の一覧中に「*」で示しています。

■　単項演算子「+ - [ ] ++ -- ¯ ! !!」

```
*     a+    ・・・・・・ a=1
*     a-    ・・・・・・ a=0
      -a    ・・・・・・ -a
*     a[    ・・・・・・ push a  (a をマクロスタックに push します)
*     a]    ・・・・・・ pop a   (a をマクロスタックから pop します)
      a++   ・・・・・・ a=a+1
      a--   ・・・・・・ a=a-1
      a¯    ・・・・・・ a=¯a
      a!    ・・・・・・ a=!a
*     a!!   ・・・・・・ a=swaphl(a)  (上位バイトと下位バイトの交換)
```

前置の単項演算子は、-（マイナス）のみです。その他は全て後置（変数の後ろ）で記述しなければなりません。そのため、

```
++i  !(fa && fb)
```

というような記述はできません。

マクロスタックは、「オプション***Bu***」で確保したマクロ配列領域の最後尾から使われます。

■　**二項演算子** 「. .. << >> & ^ | * / % + - < <= > >= == != && ^^ ||」

```
*       p.i  •••••• p[i] (char *p)  (byte ptr 演算子)
*       p..i •••••• p[i] (int  *p)  (word ptr 演算子)
```

※ i の値は、0～63までです。

※　アクセスするセグメントは、dz です。

※　その他の二項演算子は、C言語に準じます。

■　**代入演算子** 「= &= ^= |= *= /= %= += -=」

代入演算子は、C言語に準じます。

■　**演算子の優先順位**

上記の演算子は、上にあげた順に、より高い優先順位を持ちます。すなわち、通常の算術規則と異なり、「+」と「-」では「+」の方が優先順位が高くなります。これは、内部処理の簡略化の結果ですが、実用上特に弊害はありません。

システム関数

&a("<msg>")　:ask
&a(sysmsg)　(Ver1.5)

《<msg>か？（Y/N）》を問い合せます。
また、文字列の代わりにシステムメッセージNo.を与えると、VZの内部メッセージを表示します。

[返り値]

```
r == 1  : 'Y'
r == 0  : 'N'
r == -1 : [ESC]
```

[例]

```
&a("編集テキストを放棄します")
```

&b(time)　:beep

一定時間ビープ音を鳴らします。

[例]

```
&b(60)                  ; 約1秒間ビープ音を鳴らします。
```

&c　:char

キーボードから1文字入力します。

[例]

```
:A (s)?{ &c >A }; スクリーンエディットに戻るまでキー入力します。
```

&d　:display

「silentフラグ」をOFFし、全画面を再表示します。

&e(ptr)　:extern

「ptr」が示す文字列を、マクロコードとしてコールします。

[例]

```
&g("【マクロ文】") &e(pw)
```

&f(c,str)　　:find

文字「c」を文字列「str」内でサーチし、見つかった位置を返します。
[返り値]
　　r >= 0　: 見つかった位置までの文字数
　　r == -1 : 見つからない
[例]
　　&f(cd,"{}()[]<>")

&g("<title>") :get string

「<title>」ウィンドウを開き、文字列を入力します。入力した文字列は、「変数pw」が示すワークバッファに格納します。入力可能な文字列の長さは、63文字です。
また、入力した文字列が定数ならば、その値を返します。
[返り値]
　　r == 0　: 入力した文字列は、定数ではない
　　r == -1 : [ESC]
　　r == -2 : [↵]のみ　(Ver1.5)
　　上記以外: 入力した文字列が表す数値
[例]
　　&g("文字色") an=r, &d

&h　　　　:hexa

カーソル位置以降の文字列を16進数とみなして読み込み、その値を返します。
[返り値]
　　r == -1 : 16進数ではない

&i(n)　　　:interrupt

「int n」(内部割り込みn番)を実行します。受渡しレジスタとして、「変数ax, bx, cx, dx, si, di」を参照します。また、「CS, DS, ES」にはVZのコードセグメントをセットします。
[例]
　　ax=$2C00, &i($21) rn=cx+dx,

&k　　　　　:keyin

BIOSのキー入力 (AH=05h int18h) をコールします。キー入力があった場合は、最初のキーを取り込み、残りは捨てます。
[返り値]

r == 0 : キー入力なし
r != 0 : 入力キーコード
(BIOSからの返り値をAXへ、AL!=0の場合はAH=0とする)

&l　　　　　:line in

[↵]または[ESC]が入力されるまで、キーボードから入力します。
[例]

```
#45 &o('0'+r) &l
```

&m("<format>",par1,par2..) :message

---

printf文に準じた書式制御で、画面最下行へメッセージを出力します。
<format>の書式制御文字は、次のとおりです。

%[-][0][n]{d|u|x|c|s}

[例]

```
&m("Result: %d %u $%x '%c' $"%16s$"",r,r,r,r,r)
```

&?("<format>",par1,par2..) :print

---

printf文に準じた書式制御で、カーソル位置へ出力します。

&n　　　　　:numeric

カーソル位置以降の文字列を10進数とみなして読み込み、その値を返します。
[返り値]

r == -1 : 10進数ではない

&o(code)　　　:output

「code」をキーボードからの入力文字またはコマンドとして入力します。

これは、&p関数からの入力に対応します。

&p　　　　　:pause

キーボードから文字またはコマンドを入力し、そのコードを返します。
[返り値]
　　r < $FF　 : 入力文字コード
　　r >= $100 : コマンドコード（コマンド番号+256、#nn 定数に対応）
[例]
　　(-1){ &p (r > $FF)? &o }　; 文字の入力のみ無視する。

&q　　　　　:quit

サブルーチン（マクロ）コールのネストを1レベルスキップします。

&r　　　　　:random

乱数を発生させ、その値を返します。
[返り値]
　　r = rand()

&s　　　　　:silent

「silentフラグ」をONし、それ以降の画面出力を停止します。
(&m(...)によるメッセージは表示されます。)

&t　　　　　:trace

マクロの実行を中断し、トレースモードに入ります。ファンクションキー表示の左端に、中断した位置のマクロコードの一部を表示し、次のキーの入力待ちになります。
　　[**SPACE**] ………… ステップ実行
　　[**↵**] ……………… マクロの連続実行を再開
　　[**ESC**] …………… マクロ実行の中止
[**STOP**]キー（J-3100版、IBMPC版では[**Ctrl**]+[**Break**]）を押してもトレースモードに入ります。

&u(c)　　:upper

文字「c」が小文字ならば大文字に変換し、その値を返します。
[返り値]

```
r = toupper(c)
```

&v(mode)　　:video　　(Ver1.5)

カーソル表示をON／OFFします。

0 = OFF
1 = ON

&w(time)　　:wait

一定時間ウェイトします。
[例]

```
&w(60)                  ; 約1秒間ウェイトします。
```

&x(x)　　:x locate

カーソルを「xカラム」へ移動します。

&y(y)　　:y locate

カーソルを画面の上端から「Y行目」に移動します。

コマンド関数

&#35(n) &#r(n)　　:ページアップ

「n」行スクロールアップします。

&#36(n) &#c(n)　　:ページダウン

「n」行スクロールダウンします。

&#43(n) &#U(n)　　:直前のカーソル位置

「マーク#n」へカーソルを移動します。(n=0～4)

&#44(n) &#M(n)　　:カーソル位置マーク

カーソル位置を「マーク#n」に記憶します。(n=0～4)

&#45(n) &#J(n)　　:指定行番号ジャンプ

「行番号n」へカーソルを移動します。

&#63(n) &#T(n)　　:編集テキスト切替え

編集テキストを「n番」に切り替えます。

&#64(n) &#Y(n)　　:分割モード切替え

画面の分割モードを変更します。

0 = フル
2 = 横分割
4 = 縦分割

&#68(n)　　　　　　　:スペース・タブの挿入

カーソル位置から「カラムn」までの間を半角スペースまたはタブで埋めます。タブを使わない時は、マクロの中で「オプションEu」を「+」にしてください。この関数は矩形ブロックやセンタリングのマクロで使用しています。

#?

「silentフラグ」をOFFし、カーソル行を再表示します。

#]

ステータスラインを再表示します。

#03 #[　　　　　　　　　　(Ver1.5)

エディタを強制終了します。

システム変数

以下に、演算式で使用できるシステム変数の一覧を示します。

・「*」がついているものは、実行中に変更可能です。
・「1.5」は、Ver1.5で追加／変更されたものです。

■ グローバル変数

変数		内容
cm	*	直前に実行したエディットコマンド番号
ef	1.5	EMSのフリーページ数
ep	1.5	編集モード時の[ESC]の扱い ***ep***⁻　:プリフィックスキー ***ep***⁺　:エスケープキー
ks		シフトキーフラグ (PC-9801) +1:[**SHIFT**] +2:[**CAPS**] +4[ｶﾅ] +8:[**GRPH**] +16:[**CTRL**] (J-3100, IBM PC) +3:[**Shift**] +4:[**Ctrl**] +8[**Alt**]
r	*	コマンド、マクロ、メニュー等の返り値
rn	*	乱数用ワーク（seed)
s		システムモード 0 = 編集モード 1 = 文字列入力モード 2 = コマンドラインモード 3 = ファイラーモード 4 = メニュー，(Y/N)等の1文字入力モード
wc		編集中のテキストの数（コンソールファイルは除く）

■ 編集テキスト変数

変数		内容
cd		カーソル位置の文字コード
ck		カーソル位置の文字種 0 = 半角

		1 = 全角
cl		カーソル行の先頭オフセット
cp		カーソル位置のオフセット
ct		カーソル位置の文字タイプ
		0 = EOF
		1 = 改行
		2 = 00h〜20h、全角スペース
		3 = 半角記号・全角記号
		4 = ひらがな
		5 = 上記以外の全角文字
		6 = 上記以外の半角文字
fe		編集テキストの拡張子タイプ (デフォルト拡張子の指定番号)
ht	*	編集テキストのタブカラム数
kx	*	保存したカーソルのカラム位置
ky	*	保存したカーソルのウィンドウ内の行（Y）位置
ld		表示行番号 (表示行番号モードでない場合は不定)
lh		カーソル位置の論理行の先頭からのＹ位置
ln		論理行番号
lx	*	表示カーソルのカラム位置
ly		表示カーソルのウィンドウ内のＹ位置
mb	*	ブロックモード
		0 = 非ブロックモード
		1 = 行境界のブロックモード
		2 = 文字境界のブロックモード
		4 = 矩形ブロックモード
mi	*	挿入モード
		0 = 挿入モード（Insert）
		1 = 上書モード（Overwrite）
mp	*	ページングモード
		0 = ［P］表示ページングモード
		1 = ［C］論理ページングモード
		2 = ［S］文字列検索モード
mr	*	テキスト修正フラグ
		0 = 未修正
		1 = 修正あり

		-1 = Read Only
		$80 = View mode
we	1.5	編集テキストの表示カラム数
wn		編集テキスト番号（ファイル名の前の数字）
wt		画面分割モード
		0 = 全画面
		2 = 横分割の上
		3 = 横分割の下
		4 = 縦分割の左
		5 = 縦分割の右
wx		編集ウインドウの横幅
wy		編集ウインドウの高さ
xb		ブロックマーカーのカラム位置

■ ポインタ変数

変数		内容
pa		アプリケーションヒストリーバッファ
pc		VZの起動コマンド名
pd		カレントディレクトリ名
pf		ファイル名ヒストリーバッファ
pi		DEFファイルの参照パス名
pn		カレントドライブ名
pq		テンポラリファイルの参照パス名
ps		検索文字列ヒストリーバッファ
pu		マクロ配列領域の先頭
pw		ワーク用ヒストリーバッファ
px		コマンドラインヒストリーバッファ
pz		コマンド起動パス名
sp	*	マクロスタックポインタ
vp	* 1.5	画面出力制御パラメータブロック　※1

■ セグメント変数

変数		内容
cz		コードセグメント
dz	*	データセグメント　※2
gz	1.5	テキスト管理セグメント　※3
tz		編集テキストバッファのセグメント
kz		テキストスタックのセグメント
kp		テキストスタックエンドオフセット

※1 次の構造体へのポインタです

+0	far ptr dsp	カーソル位置の絶対アドレス（offset, segment）
+4	char win[4]	表示ウィンドウ（開始x, y, サイズx, y）
+8	char loc[2]	カーソル位置 x, y
+10	char dosloc[2]	DOSのカーソル位置 x, y
+12	char dspatr	表示属性

※2 ポインタ演算の際にこのセグメントを使います。
※3 ラインバッファ、編集テキストワークのセグメントで、SSの値です。

マクロに関する特記事項

■　[返り値] r について

^KO マクロが示すように、<prefix>[0]～[9] キーを入力すると、<prefix>0 のマクロをただちに実行します。その時、変数「*r*」には、[0]～[9]のキーに対応して 0～9 の値をセットします。

その他の場合のマクロ呼び出し時の「*r*」には、直前に実行したコマンドの返り値をそのまま保存しています。

ローカルマクロをコールした場合、最後に(式)で評価した値を変数「*r*」にセットして、コール元のマクロに戻ります。

■　マクロ呼び出し時の内部レジスタの値

マクロ呼び出し時の内部レジスタには、変数 **s**（システムモード）を代入します。従って、編集モード時のみ動作させるマクロは、その先頭で、「(**s**)?.」とする代わりに、「?.」と省略して記述することができます。

■　マクロキーとコマンドキーの優先順位（Ver1.5）

従来は、常にマクロキーの方が優先順位が高かったため、例えば[SHIFT]+[←]にマクロを割り当てようとすると、編集モード以外でこれを使用しているコマンドが動作しなくなりました。このため、「mi.def」などでは

```
30 ¥[←] "画面の左端"
        (s)>?{ { &x(0). } #11 #11 #88 }
```

というように、システムモード（**s**）を調べて、編集モード以外ではそれぞれのコマンドを実行するように記述しています。Ver1.5では、この問題に対処するために、

「編集モード以外ではコマンドキーを先にチェックする」

ように変更しました。従って、上記マクロは、

```
30 ¥[<--] "画面の左端"
        &x(0)
```

というように簡潔に記述できます。

■ [ESC]キーの利用（Ver1.5）

編集モードでは、通常[**ESC**]キーはプリフィックスキーに割り当てられているため、例えばバックグラウンドで動作するマクロの実行を[**ESC**]キーで中断することができませんでした。このため、Ver1.5で「オプション***Ep***」を追加しました。

EP-　:[**ESC**]キーはプリフィックスキー（デフォルト）

EP+　:[**ESC**]キーをそのまま入力

マクロ中で「***Ep+***」に設定すれば、[**ESC**]キーでマクロを終了できます。

付　録

A.コマンド一覧

B.オプション一覧

C.エラーメッセージ

A.コマンド一覧 （PC-9801版のみ）

■ 共通コマンド

#	標準キー		機能略説
01	^[		直前のコマンド実行
02	^_		キーボードマクロ
03		[ESC]	[エスケープ
04	^M	[↵]	m リターン
05	^E	[↑]	e ↑ (A) ※
06	^X	[↓]	x ↓ (B)
07	^S	[←]	s ← (A)
08	^D	[→]	d → (B)

■ ラインエディット

#	標準キー		機能略説
09	^A	[SHIFT]+[←]	a 1語← (A)
10	^F	[SHIFT]+[→]	f 1語→ (B)
11	^QS	[CTRL]+[←]	< 論理行頭
12	^QD	[CTRL]+[→]	> 論理行末
13	^V	[INS]	v 挿入モード
14	^H	[BS]	h バックスペース (A)
15	^G	[DEL]	g 1文字削除 (B)
16	^QH	[SHIFT]+[BS]	語の先頭まで削除 (A)
17	^T	[SHIFT]+[DEL]	t 語の終りまで削除(B)
18	^QT	[CTRL]+[BS]	行頭まで削除
19	^QY	[CTRL]+[DEL]	l 行末まで削除
20	^U		u 削除文字（列）復活
21	^I	[TAB]	タブ
22	^P		制御文字入力
23	^QU		1語大／小文字変換
24	^QI	[SHIFT]+[f・5]	I 文字列の複写
25	^QN		P ファイル名の複写

■ カーソルの移動

#	標準キー		機能略説		
30	^@	[SHIFT]+[INS]	@	ページングモード	
31	^W	[SHIFT]+[ROLL DOWN]	w	ロールアップ	(A)
32	^Z	[SHIFT]+[ROLL UP]	z	ロールダウン	(B)
* 33				ロールアップ2行	(A)
* 34				ロールダウン2行	(B)
35	^R	[SHIFT]+[↑]	r	ページアップ	(C)
36	^C	[SHIFT]+[↓]	c	ページダウン	(D)
37	^QW	[ROLL DOWN]		スムースアップ	
38	^QZ	[ROLL UP]		スムースダウン	
39	^QE	[CTRL]+[↑]		画面の上端	
40	^QX	[CTRL]+[↓]		画面の下端	
* 41	^QR	[CTRL]+[ROLL DOWN]	^	ファイルの先頭	
* 42	^QC	[CTRL]+[ROLL UP]	_	ファイルの最後	
43	^QP		U	直前のカーソル位置	
44	^QM		M	カーソル位置マーク	
45	^QJ	[SHIFT]+[f•2] J	J	指定行番号ジャンプ	

■ 行・ブロックの操作

#	標準キー		機能略説		
46	^B	[f•10]	b	ブロックモード	
47	^Y	[f•8]	y	行・ブロック削除	
48	^J	[f•9]	i	行・ブロック挿入	(E)
49	^KK	[SHIFT]+[f•8]	k	行・ブロック記憶	
50	^KC	[SHIFT]+[f•9]	p	行・ブロック複写	(E)
51	^KY	[SHIFT]+[f•2] C		スタックの消去	
52	^QB	[SHIFT]+[f•2] B	B	ブロックの先頭／最後	
53	^N	[SHIFT]+[↵]	n	1行挿入	
54	^QL	[SHIFT]+[f•2] U		行の回復	

■ 文字列の操作

#	標準キー	機能略説	
55	^QF [f･6]	F 検索文字列の設定	
56	^QA [f･7]	文字列の置換	(F)
57	^QO	直前の置換の再実行	(F)
58	^QK [SHIFT]+[CLR]	対応する()[]{}<>	
59	^L [f･5]	G 検索文字列の取得	(G)
60	^KG	タグ情報の取得	
61	^KA [SHIFT]+[f･7]	文字列の選択複写	

■ ウィンドウ･マルチファイルの操作

#	標準キー	機能略説
62	^KW [f･2]	W ウィンドウ切替え
63	^KT [f･3]	T 編集テキスト切替え
64	^KS [f･4]	Y 分割モード切替え
65	^KF [SHIFT]+[f･4]	分割位置の変更
66	^KL [SHIFT]+[f･2] L	表示行数切替え
67	^KR [SHIFT]+[f･2] R	Read onlyモード
68	^KH [SHIFT]+[f･2] I	インデントの変更
69	^KX [SHIFT]+[f･3]	テキストの比較

■ ファイル関連の操作

#	標準キー	機能略説	
70	[ESC] O	O ファイルのオープン	(H)
71	[ESC] R	R Read onlyオープン	(H)
72	[ESC] N	N 新規ファイルオープン	
73	[ESC] C	C ファイルのクローズ	
74	[ESC] L	L クローズ･オープン	
75	[ESC] S	S ファイルのセーブ	
76	[ESC] A	A 既存ファイルへ追加	
77	[ESC] Q	Q エディタを終了	
78	[ESC] E	E DOSコマンド実行	

	79	［SHIFT］+［ESC］	X DOS画面出力の参照
	80	^K^	再カスタマイズ
	81	^K_	キーボードマクロ文挿入
	82	［ESC］P	ファイル名の変更 （I）
	83	［ESC］W	プロファイル情報のセーブ

■ ファイラー

	#	標準キー	機能略説
*	88	［SHIFT］+［←］	選択←
*	89	［SHIFT］+［→］	選択→
*	90	［SHIFT］+［↑］	選択↑
*	91	［SHIFT］+［↓］	選択↓
*	92	［BS］	先頭／最後
*	93	［TAB］	ウィンドウ切替え

■ DOSコマンドライン

#	標準キー	機能略説
95	［SHIFT］+［ESC］	DOS画面出力の参照
96	^W	エイリアス一覧
97	［CTRL］+［↵］	コマンドの展開再表示
98	［SHIFT］+［↵］	ファイラー

■ マクロコマンド

&	標準キー	機能略説
01	［HELP］	ヘルプ
02	［f・1］	ファイルメニュー
03	［SHIFT］+［f・1］	モードメニュー
04	［SHIFT］+［f・2］	サブメニュー
05	［ESC］M	マクロメニュー
06	［ESC］D	テキストの二重化
07	［ESC］U	編集のやり直し

	08 [ESC] I	カーソル位置へ挿入
	09 [ESC] X	全ファイルのクローズ
	10 [ESC] ^	外部マクロの実行
	11 [ESC] @	マクロ文の実行
	12 ^Q[	表示行頭
	13 ^Q]	表示行末
	14 [CLR]	表示位置変更
	15 ^KI	タブの切替え
	16 ^K1〜4	マーク#1〜#4
	17 ^Q1〜4	マーク#1〜#4へジャンプ
	18 [SHIFT]+[f・10]	タグジャンプ
	19 ^KD	カーソル行の複写
	20 ^KU	ペーストのアンドゥ
	21 [SHIFT]+[f・6]	複数テキストで検索・置換
	22 [CTRL]+[TAB]	文字列の補完
*	23 ^KP	プリント専科IIの起動
*	24 ^L [F05]	検索文字列の連続取得"
*	25 ^KJ	文字コード表示
*	26 ^QV	View モード
*	27 [GRPH]+[X]	強制終了
(+block. def)		
*	60 ^KB	矩形ブロックモード
	65 [ESC]]	行末のSPACE・TABの削除
(+zenhan. def)		
*	69 ^KZ	全角／半角変換
(+keisen. def)		
*	70 [ESC] H	作表メニュー
(+tool. def)		
*	70 [ESC] H	ツールメニュー

※「*」は Ver1.5 の追加・変更コマンド

■　コマンド実行後の返り値r

(A)　-1 = ファイルの先頭

(B)　-1 =ファイルの最後

(C),(D)　・文字列検索モード

　　0 = 見つかった

　　-1 = 見つからない

　・上記モード以外

　　(A),(B)と同じ

(E)　ペーストした行•ブロックのタイプ

　0 = 行

　1 = 行単位のブロック

　2 = 文字単位のブロック

(F)　置換した文字列の数

(G)　取得した文字列の長さ

(H)　0 = ファイルをオープン

　1〜 既存のテキスト

(I)　0 = ファイルは存在

　2 = ファイルは存在しない

　3 = 無効なパス名

　4 = アクセス不能

　5 = リードオンリー

　-1 = ハードエラー

(その他)

　0 = 正常終了

　-1 = 異常終了（エラー、キャンセル）

B.オプション一覧

■ 特殊オプション

記号	内容
z	VZを常駐または解放します。
w	DEFファイルを vz.com 本体に組み込みます。
r	起動後ただちにマクロを実行します。

■ メモリ／バッファサイズ

記号	標準設定	内容
Bt	64(KB)	テキストバッファのサイズ
Bq	0 (KB)	メモリ上のテンポラリバッファのサイズ
Bf	32(KB)	DOSコマンド用メモリの最小値
Bo	32(KB)	コンソール出力取込みバッファサイズ
Bm	1024	マクロバッファのフリーエリア
Ba	0	エイリアスバッファ 〃
Hs	256	文字列ヒストリー 〃
Hf	128	ファイル名 〃
Hx	128	DOSコマンドライン 〃
Ha	64	アプリケーション 〃
Bl	1024	行バッファサイズ
Bc	256	削除文字列バッファサイズ
Bu	64	マクロ配列領域サイズ

■ 各種モード

記号	標準設定	内容
GV	0	裏VRAM使用モード
＊***EM***	+	EMSの使用ページ数
＊***SW***	1	スワップモード
TC	10	オープン可能なテキスト数
＊***WD***	80	デフォルトの表示カラム数（編集ファイルは WE）

PG	0	表示行番号のページ数
RS	3	スムーススクロール速度
Ta	8	タブカラム数
Tb	4	.C .H のタブカラム数
Ci	4	カーソル形状（挿入）
Co	7	〃　（上書）
FW	256	ディレクトリバッファの数
FV	1	ディレクトリ表示モード
FS	0	ディレクトリソートオプション
FO	3	ディレクトリコンペアオプション
WL	0	ウィンドウの表示位置
LC	-	エディット時の画面行数

■ 表示属性

記号	標準設定	内容
An	7	テキスト
Ac	5	コントロール文字
Al	5	行番号
Ah	13	ブロック
As	15	ステータスバー
Ao	14	コンソールファイル
Ar	13	Read onlyファイル
Am	6	メッセージ
Ab	5	ウィンドウわく
At	5	ウィンドウタイトル
Aw	7	ウィンドウ文字
Af	6	ウィンドウ先頭文字
Ai	15	ウィンドウカーソル
Aj	6	ヘルプウィンドウ
Ak	6	キーモード
Ag	1	ファイラー背景
Au	6	ファイラーカーソル
* *Ap*		ファンクションキー（J-3100,IBM PC版）

■ スイッチ（表示関係）

記号	標準設定	内容
Dl	-	行番号の表示
Dc	+	CR の表示
Dt	-	TABの表示
De	+	EOFの表示
Ds	-	システム情報の表示
Db	-	¥文字の表示
Dn	-	論理(-)／表示(+)行番号
Du	-	カーソル行アンダーライン
Df	-	フルパス名で表示
Dp	-	ファイル名の小文字表示

■ スイッチ（編集関係）

記号	標準設定	内容
Ei	+	オートインデント
Es	+	^QE, ^QX でスムーススクロール
Ez	+	EOFコードの付加
Eb	+	バックアップファイルの作成
Eu	-	[TAB]でスペースコードを入力
Et	+	タイトルサーチ
Eh	-	全(-)／半(+)画面ページング
Ew	+	ワードサーチ
Ec	-	検索時の英大／小文字の同一視
Ex	-	テキストスタックフル時の警告
En	-	カーソルを行末以降に表示させない
Ev	-	ジャンプ後カーソルを画面の中央へ
Ek	-	ブロックペースト後カーソル位置更新
* ***Ea***	-	表示カラム数の自動設定機能
* ***RO***	-	View モード

■ スイッチ（その他）

記号	標準設定	内容
Sx	+	コンソール出力の取込み
Se	+	エスケープシーケンスのスキップ
Sd	+	DOSコマンドラインの展開再表示
Sk	-	DOSコマンド実行前に、テキストスタックをクリア
Sr	-	editfile にプロファイル情報をセーブ
Sy	-	[↵]/[SPACE] で [Y]/[N]
St	-	DEFファイルのタイトル表示
Sn	+	新規ファイルの確認
Sq	-	[ESC] Q, [ESC] C で即座に終了
FP	+	FEPを制御
FR	-	日本語入力モードで起動
FT	0	FEPのタイプ(0=自動識別, 1=vjeb, 2=mttk, 3=atok)
* *FK*	-	拡張ファンクションキー

※ 「*」はVer1.5の追加・変更オプション

※ 標準設定値はvz. defの値

C.エラーメッセージ

■ 起動時のエラー

カスタマイズエラー（CTRL-Cで中止）

DEFファイルにエラーがあります。

[CTRL]+[C]　　：中断
その他のキー　　：強行

Ver1.0のVZでVer1.5のDEFファイルを読むとエラーになります。

"・・・・"が見つかりません.

指定のパスに、DEFファイルが見つかりません。環境変数「VZDEF」の設定をご確認ください。

メモリが足りません.

DEFファイルのバッファサイズ指定オプションでバッファを大きく取りすぎています。いずれかの値を減らしてください。また、たくさんのマクロを記述したDEFファイルを組み込む際にも発生する場合があります。

Ver1.5は、コードサイズが5KBほど増えているため、起動時に一度に読み込めるDEFファイルのサイズが、その分小さくなっています。このような場合は、DEFファイルをいくつかのファイルに分割し、インクルードしてください。

■ 編集時のエラー

"・・・・"は入力できません

ファイル入力時に、なんらかのエラーが発生して入力できません。

"・・・・"は出力できません

指定パス名が無効なパス名、またはリードオンリーで、出力できません。

テンポラリファイルエラー

テンポラリファイルを作成しているディスクがいっぱいになったか、またはテンポラリファイルになんらかの異常が発生しました。これ以上、編集を続けることができません。ただちに編集を中断して原因を調べてください。環境変数「TMP」が設定されていない場合、テンポラリファイルは起動時のカレントディレクトリに作成します。この環境

変数は、VZの起動時にしか参照しませんので注意してください。

大きなファイルを編集する際には、テンポラリファイルを作成しないViewモードでオープンすることもできます。

ラインバッファがいっぱいです

一行の長さが長すぎて、入力・編集できません。[↵]で2行に分割してください。どうしても長い行を作りたい場合は、いったん終了し、「オプション*Bl*」の値を適当に増やして、再度起動してください。

オープンできません

すでに「オプション*TC*」で指定したオープン可能なテキスト数（標準設定で10個）のテキストを編集しており、これ以上オープンできません。不用なテキストを[**ESC**] **C** コマンドでクローズして、再度オープンしてください。

ディスクがいっぱいです

出力するディスクがいっぱいで、最後まで出力できませんでした。不用なファイルを消去するか、セーブ先を変更して、再度セーブしてください。

テキスト領域がいっぱいです

コンソールファイルに対してサイズを増やす編集操作を行ないましたが、これ以上サイズを増やせません。

ブロックが大きすぎます

ブロックが64KBを越えています。64KBにおさまる範囲のみ処理し、残りの部分をブロックとして残しました。

メモリが足りません

あらたにテキストをオープンしようとしましたが、十分なメモリがありません。編集中の他のテキストのうち、サイズの大きないずれかのテキストをクローズしてください。起動時に「オプション*Bt*」を「16」に設定してテキストバッファを小さくすると、処理速度は遅くなりますが、メモリ消費は少なくなります。

常駐時にこのエラーが発生した場合、VZの編集テキストの上に、さらに別のプログラムが常駐していることが考えられます。

バッファがいっぱいです

DEFファイルを再カスタマイズしたか、あるいは外部マクロを実行しようとした際、マクロバッファがいっぱいになりました。「オプション*Bm*」のマクロバッファサイズをも

っと増やして、再起動してください。

COMMAND.COMを起動できません

コマンドラインモードでDOSコマンドを実行しようとした際、指定パスに「command.com」が見つからないため、DOSコマンドを実行できません。
MS-DOSのバージョンによっては、環境領域が正常に参照できずに、このエラーが発生する場合があります。

カスタマイズエラー

DEFファイルを再カスタマイズした際、カーソル位置の行でエラーが発生しました。

■ デバイスのハードエラー

ドライブA:の書込み禁止です (Abort/Retry)?

ドライブAのディスクがライトプロテクトされており、出力できません。

ドライブA:の指定が違います (Abort/Retry)?

ドライブAは存在しません。

ドライブA:の準備ができていません (Abort/Retry)?

ドライブAにディスクがセットされていません。

ドライブA:のハードエラー#nです (Abort/Retry)?

ドライブAに、n番のハードエラーが発生しました。エラーの内容は次のとおりです。

#4 ······· データのCRCエラー
#6 ······· シークエラー
#8 ······· セクタが存在しない（フォーマットされていない）
#10 ······ 書き込み不良
#11 ······ 読み出し不良
#12 ······ 一般的なディスク不良

索引

■ 英数字・記号項目

7to5／6.com …… 2-14, 8-5
$ …… 10-13
@ …… 3-15, 3-16
ATOK …… 2-14, 8-5
block.def …… 4-24
command.com …… 2-6, 6-3, 7-2
CONSOLE …… 6-11
DEF ファイル …… 2-2, 10-2
DEF ファイルの書式 …… 10-9
DEF ファイルの取り込み …… 10-6
DEF ファイルの分割 …… 10-3
DEF ファイルの読み込みパス …… 2-4
DOS コマンドの実行 …… 1-32, 6-2
editfile …… 3-15, 10-18
EMS …… 2-13, 7-7
ezkey.com …… 3-2, 8-8, 10-11
Edit モード …… 3-11
FEP の制御 …… 2-14, 9-7
fc …… 5-15
froll.com …… 8-3
I／O バンク RAM …… 2-14
IBM PC／AX 版 …… 8-6
J-3100 版 …… 8-3
LASTDRIVE …… 5-7
lharc …… 5-15, 5-17
MAKE ユーティリティ …… 9-5
PATH …… 2-6
printf …… 11-13
prompt …… 6-12
RAM ディスク …… 2-12
Readonly …… 1-30, 3-4, 3-10
sample1.txt …… 1-2
SET コマンド …… 2-6
start.bat …… 1-2
TMP …… 2-8
Undo …… 1-16
View モード …… 2-9, 3-5, 3-11
vmap.com …… 7-2, 7-5, 8-13
VZBAK …… 2-7, 3-7
VZDEF …… 2-7, 10-3
VZPATH …… 2-7, 2-10
VZPATH のメニュー …… 5-9
vz.bat …… 8-3
vz.def …… 10-2
vzs.com …… 2-14
vzs.def …… 10-5
vzsel.com …… 8-3
WordStar …… 1-2, 4-2

■ 和文項目

【ア行】
アセンブル・リンク …… 9-4
アセンブルスイッチ …… 9-5
インジケータ …… 1-4
インスタントマクロ …… 11-3
インデントの変更 …… 4-34
ウィンドウ切替え …… 3-13
ウィンドウ枠 …… 10-18
裏グラフィック VRAM …… 2-4, 2-13, 7-6
裏ファンクションキー表示 …… 8-11
上書きモード …… 1-11, 4-12
英大／小文字の変換 …… 4-14
エイリアス …… 6-6, 10-16
エイリアスの再定義 …… 10-8
エスケープシーケンス …… 6-12
エスケープ文字 …… 4-36
エディタを終了 …… 1-32, 3-8
演算子 …… 11-24
オートインデント …… 1-34, 4-14
オプション …… 10-16
オプション指定 …… 2-4
オプション変数 …… 11-24
大きなブロックの削除 …… 4-23

【カ行】
カーソル …… 1-5
カーソルの形状 …… 4-12, 10-20

カーソル位置の保存 …… 4-5
カーソル位置へ挿入 …… 1-31, 3-12
カーソル位置マーク …… 4-9
改行コードの検索 …… 1-18, 4-31
拡張子による指定コマンド実行 …… 6-10
拡張子の省略 …… 2-10
カスタマイズ …… 10-2
仮想ドライブ …… 5-7
カット&ペースト …… 4-20
外部マクロの実行 …… 4-24, 10-8, 11-8
画面表示の停止 …… 11-15
環境変数 …… 2-6
環境変数の参照 …… 6-5
キーボードマクロ …… 1-38, 11-2, 11-6
キーボードマクロ文挿入 …… 11-6
キーボードロック …… 8-9
既存ファイルへ追加 …… 1-30, 3-8
起動時オプション …… 10-19
基本コマンド …… 4-2, 11-19
強制スワップ指定 …… 7-11
強制終了 …… 3-9
行の回復 …… 4-33
行番号ジャンプ …… 4-8
行番号モード …… 1-34, 4-9
行末以降のカーソル表示 …… 4-5
行末のスペースの削除 …… 4-25
矩形ブロック操作 …… 4-24
クリック音 …… 8-11
クローズ・オープン …… 1-30, 3-6
グラフキーの利用 …… 9-9
グローバルマクロ …… 11-20
罫線文字入力 …… 4-36
検索 …… 1-17, 4-26
検索・置換の中断 …… 4-28
検索時の英大／小文字の識別 …… 1-34, 4-32
検索文字列の取得 …… 4-29
検索文字列の設定 …… 4-26
検索文字列バッファ …… 4-30
検索モード …… 4-26, 4-6
コマンドキー …… 10-10
コマンドライン …… 2-2
コマンドラインモード …… 6-2
コマンド関数 …… 11-20, 11-31
コンソールバッファ …… 7-6
コンソールファイル …… 6-11
コンソール出力の取り込み …… 6-11

【サ行】

再カスタマイズ …… 6-8, 10-7, 11-9
最後のテキストをクローズ …… 3-6
削除 …… 1-11, 1-16, 4-12
削除文字列の復活 …… 4-13
サブメニュー …… 7-9
サブルーチン …… 10-16
システムメニュー …… 5-8, 5-18, 6-10, 10-14
システム関数 …… 11-20, 11-24, 11-26, 11-33
ジャンプ …… 4-8
修正フラグ …… 3-5
修正モード …… 1-4
常駐時のメモリ消費 …… 7-5
常駐モード …… 1-36, 2-12, 6-2, 7-2
新規ファイル …… 1-30, 3-3, 3-5
スイッチ …… 10-24
スクリーンエディット …… 4-2
スタックの消去 …… 4-22
ステータスライン …… 1-4
ステータスラインが水色 …… 3-4
ステータスラインが黄色 …… 6-11
スムーススクロール …… 1-15, 1-28, 4-4, 4-7
スワップ …… 7-10
制御文字入力 …… 4-36
センスモード …… 4-8
全ファイルのクローズ …… 1-32, 3-12
ソースコード …… 9-2
挿入モード …… 1-11, 4-12

【タ行】

対応するカッコの検索 …… 4-28
タイトル行 …… 1-20, 4-7
タイトル行サーチ …… 4-6, 10-18
タグジャンプ …… 4-10
タグ情報の取得 …… 4-10

タブ……4-13
タブの切り替え……4-33
単語の区切り……1-17, 4-15
置換……1-9, 4-27
置換の再実行……4-28
直前のカーソル位置……4-9
直前のコマンドの実行……4-33
テキストのバイト数……3-13
テキストの二重化……1-31, 3-11
テキストの比較……4-34
テキストウィンドウ……10-12
テキストスタック……4-21
テキストスタックのサイズ……4-23
テキストバッファのサイズ……3-4
テンプレート機能……1-32, 7-5
テンポラリファイル……2-8, 7-11
テンポラリファイルエラー……2-8
ディスクの空き領域……2-13, 5-3
ディレクトリの再読み込み……5-6
ディレクトリを省略……2-7, 2-10
デフォルトバイナリー拡張子……2-10, 10-18
デフォルト拡張子……2-10, 5-8, 10-18
動作メモリが限定……2-14
動作環境の設定……2-12
動作中のオプション変更……2-5
同時にオープン可能なファイル数……3-4
特殊文字の入力……10-18
トレースモード……11-16

【ナ行】

入力できないキー……10-11
入力ファイルウィンドウ……2-2, 3-12, 3-3, 7-4

【ハ行】

ハードウェアスクロール……8-3
ハードディスク……2-13
バックアップディレクトリ……2-7, 3-8
バックアップファイル……1-34, 3-7
バックアップモード……2-7, 2-9
バッククォート文字……8-11
バックスペース……4-12
バッファリング……6-13
パック……5-17
パラメータの参照……6-7
ヒストリー……10-17
ヒストリーのワード検索……4-17
ヒストリー機能……4-16
ヒストリーバッファの初期文字列……4-18, 10-17
ピックファイル……3-16
表示ページングモード……4-6
表示行……1-34, 4-3
表示行番号モード……4-9
表示行数の切り替え……3-14, 4-34
表示属性……10-23
ファイラー……1-35, 5-2
ファイルのオープン……3-3
ファイルのクローズ……1-29, 3-5
ファイルのセーブ……1-29, 3-6
ファイルの比較……1-27
ファイルメニュー……1-25, 1-29, 3-2
ファイル名の小文字表示……3-4
ファイル名の表示……3-4
ファイル名の複写……4-14
ファイル名の変更……1-31, 3-10
ファンクションキー……3-13, 4-20, 4-26, 10-12
フルパス名……3-4, 3-10
フロッピーディスクのみ……2-12
ブルーバック……8-10
ブレークポイント……11-17
ブロック操作……4-21
ブロックの先頭／最後……4-23
ブロックモード……1-4, 3-7, 4-20
ブロック指定……1-23
ブロック内の置換……1-24
プリフィックスキー……10-11
プリント専科の起動……4-35
プロファイル機能……3-15
プロファイル情報のセーブ……3-6, 3-9, 3-9, 3-12
プロファイル情報の書き出し……3-15
プロファイル情報の読み込み……3-16
プロンプト……6-2, 10-18
複数のDEFファイル……10-5

複数テキストの検索・置換 …… 4-28
複数ファイルのオープン …… 1-25, 4-4
複数ファイルのコピー …… 5-16
分割モード切替え …… 3-14
分割位置の変更 …… 3-14
ページフレーム …… 7-7
ページングの単位 …… 4-4
ページングモード …… 1-4, 4-6, 4-26
ページング単位 …… 1-34
ペーストのアンドゥ …… 4-22
変数 …… 11-23
編集のやり直し …… 1-31, 3-11
編集テキスト切替え …… 3-13
編集ファイル指定 …… 2-3
編集ファイル名の省略 …… 2-10, 3-8
ボールド …… 2-2, 8-5
ポインタ変数 …… 11-13, 11-35
ポップアップメニュー …… 11-19

【マ行】

マーク …… 4-8, 4-10
マーク位置へジャンプ …… 4-10
マクロ …… 10-15, 11-2
マクロのキーワード …… 11-19
マクロのコーディング …… 11-11
マクロのコール …… 11-20
マクロの特記事項 …… 11-37
マクロの式 …… 11-23
マクロの中断 …… 11-16
マクロファイルの書式 …… 11-11
マクロファイルの分割 …… 11-9
マクロメニュー …… 11-7
マクロ言語仕様 …… 11-18
マルチ …… 4-28
マルチステートメント …… 6-5
マルチファイル …… 1-26
マルチファイル …… 3-13
マルチ機種対応 …… 8-2
メディアの交換 …… 5-6
メニュー …… 10-14
メニューコマンド …… 5-7
メモリ・バッファサイズ指定 …… 10-20
メモリ消費 …… 10-12
メモリ上のテンポラリファイル …… 2-8, 2-12
モードメニュー …… 1-33
文字のタイプ …… 4-15
文字コードの取得 …… 11-14
文字コード表示 …… 4-35
文字列の複写 …… 4-29
文字列の補完 …… 4-30
文字列内の制御文字 …… 10-13
文字列入力モード …… 4-16

【ラ行】

ラッチモード …… 4-8
リードオンリ …… 3-4
リードオンリ属性のファイル …… 3-11
リセット …… 8-9
リターン …… 4-14
ローカルマクロ …… 10-16, 11-20
論理ページングモード …… 4-6
論理行 …… 1-34, 4-3
論理行番号モード …… 4-9

【ワ行】

ワードサーチ …… 4-31
ワードスタータイプ …… 1-5
割り込みベクタ …… 9-6

■ コマンドキー

^A …… 1-7, 4-2
^B …… 1-22
^C …… 1-10, 4-4
^D …… 1-6, 4-2
^E …… 1-6, 4-2
^F …… 1-7, 4-2
^G …… 1-11, 1-16, 4-12
^H …… 1-11, 1-16, 4-12
^J …… 1-16, 1-22, 4-21
^L …… 1-37, 4-29, 4-6
^M …… 1-10
^N …… 4-14
^P …… 4-36, 1-10
^R …… 4-4
^S …… 1-6, 4-2
^T …… 1-16, 4-12
^U …… 1-16, 4-13
^V …… 1-11, 4-12
^W …… 1-9, 4-3
^X …… 1-6, 4-2
^Y …… 1-16, 1-22, 4-21
^Z …… 1-9, 4-3
^ […… 4-33
^¥ …… 11-5
^@ …… 1-17, 1-20, 4-6

^QA …… 1-19, 1-24, 4-27
^QB …… 4-23
^QC …… 1-14, 4-5
^QD …… 1-14, 4-3
^QE …… 1-14, 4-4
^QF …… 1-17, 4-6, 4-26
^QH …… 1-16, 4-12
^QI …… 1-37, 4-29
^QJ …… 10-17, 4-8
^QK …… 4-28
^QL …… 4-33
^QM …… 4-9
^QN …… 4-14
^QO …… 4-28
^QP …… 4-9
^QR …… 1-14, 4-5
^QS …… 1-14, 4-3
^QT …… 1-16, 4-13
^QU …… 4-14
^QV …… 3-10
^QW …… 1-15, 1-28, 4-7
^QX …… 1-14, 4-4
^QY …… 1-16, 4-13
^QZ …… 1-15, 1-28, 4-7
^Q […… 1-14, 4-3
^Q] …… 1-14, 4-3
^Q1～4 …… 4-10
^KA …… 1-37, 4-29
^KB …… 4-24
^KC …… 1-23, 4-22
^KF …… 1-27, 3-14
^K …… 4-10
^KH …… 4-34
^KI …… 4-33
^KJ …… 4-35
^KK …… 1-22, 4-22
^KL …… 3-14, 4-34
^KP …… 4-35
^KR …… 1-30, 3-4, 3-10, 3-11
^KS …… 1-26, 3-14
^KT …… 1-26, 3-13
^KU …… 4-22
^KW …… 1-26, 3-13
^KX …… 1-28, 4-34
^KY …… 4-22
^K^ …… 6-8, 11-9, 11-13
^K_ …… 11-6
^K1～4 …… 4-10

ESC + A …… 1-30, 3-8
ESC + C …… 1-29, 3-5
ESC + D …… 1-31, 3-11
ESC + E …… 1-32, 6-2, 7-12
ESC + I …… 1-31, 3-12

ESC + L ……… 1-30, 3-6
ESC + N ……… 1-30, 3-5
ESC + O ……… 1-25, 3-3
ESC + P ……… 1-31, 3-10
ESC + Q ……… 1-5, 1-32, 3-8
ESC + R ……… 1-30, 3-4
ESC + S ……… 1-29, 3-6
ESC + U ……… 1-31, 3-11
ESC + W ……… 3-15
ESC + X ……… 1-32, 3-12
ESC + 英字 ……… 3-2
ESC + ^ ……… 4-24, 11-8
ESC +] ……… 4-25
ESC + @ ……… 2-5, 7-10, 8-7, 10-17, 11-3

Alt + X ……… 3-9
Alt + 英字 ……… 3-2
Ctrl + Break ……… 4-28, 6-5, 11-16
CTRL + f・n ……… 8-12
CTRL + TAB ……… 4-30
CTRL + テンキー ……… 4-36
GRPH + ……… 3-9
GRPH + 英字 ……… 3-2
HELP ……… 1-7, 11-14
NFER ……… 8-12
SHIFT + CR ……… 6-9
STOP ……… 4-28, 6-5, 11-16
TAB ……… 4-13, 5-5

■ オプション

An ……… 2-2
Ap ……… 8-4
Ba ……… 6-8, 10-7
Bc ……… 4-13
Bf ……… 2-14, 7-8
Bl ……… 2-14
Bm ……… 2-14, 10-7, 11-5
Bo ……… 2-14, 6-12, 7-6
Bq ……… 2-8, 2-12, 2-14
Bt ……… 2-8, 2-14, 3-4
Ci ……… 4-12
Co ……… 4-12
Df ……… 3-4
Dp ……… 3-4
EM ……… 7-7
Eb ……… 3-8
Ec ……… 4-32
Eh ……… 4-4, 5-6
Ei ……… 4-14
Ek ……… 4-22
En ……… 4-5
Ep ……… 11-38
Es ……… 4-4
Eu ……… 4-13, 4-25
Ew ……… 4-32
Ex ……… 4-23
FK ……… 8-6
FR ……… 2-14
FS ……… 5-10
FT ……… 2-14
FW ……… 2-14
GV ……… 2-13, 7-6
RS ……… 4-8
SW ……… 2-13, 7-10
Sd ……… 6-5
Se ……… 6-12
Sn ……… 3-3
Sq ……… 3-6, 3-9
Sr ……… 3-15
St ……… 10-5, 10-9
Sx ……… 6-12
TC ……… 2-14, 3-4
Ta ……… 4-13
Tb ……… 4-13
w ……… 10-6

VZ Editor DOS/V版(PS/55版)の配布について

VZ Editor DOS/V版の発売にともない、VZ PS/55版も同梱される事になりました。

DOS/V版(IBM日本語DOS J4.0/V以上)、PS/55のJ-DOS(IBM日本語DOS J4.0以上)、US版(英語版MS-DOS 3.3以上)が含まれております。US版は2バイトコードの処理をはずしてある為、ドイツ語等で使用する特殊文字の使用も可能になっています。

申込みは本マニュアルに挟み込んである郵便振替用紙にて受け付けます。今回のVZ DOS/V版のパッケージ内容は3.5インチ720KBシステムディスク1枚。入門マニュアル1冊です。価格は1,500円+消費税45円+送料·手数料255円の合計1,800円になります。

なお、シリアルナンバー確認の必要から、上記方法以外での扱いはいたしません。

■対応機種■

DOS/V動作機(IBM-AT互換機)および5530Z/S/T, 5551-S/T/V, 5571-S/T/V

VZ Editor FM対応版の配布について

徳田楠男(k. tok)氏作のVZ Editor Version 1.5 対応のFM版は、PDSとして日経MIX、NIFTY-Serveをはじめとしたネットワークに掲載し、またビレッジセンターからも配布しています。5インチあるいは3.5インチの2HDのシステムディスク1枚での配布です。このFM版は、ビレッジセンターでは、一切のサポートはいたしません。利用を希望される方はノーサポート·了解した上でお申込みください。価格は1,500円+消費税45円+送料175円の合計1,720円なります。専用の振替用紙を挟み込みしてありますのでご利用ください。

■対応機種■

FM16 β, FMR10/30/50/60/70, TOWNS(MS-DOS)

VZ Editor Version 1.5 ユーザーズマニュアル

1992年12月7日　第3版1刷

発行者　中村満
発行所　株式会社ビレッジセンター
〒182 東京都調布市富士見町2-2-12
電話 0424-88-3381

※本製品・マニュアルの一部または、全部を当社の許可なく複写、複製することを禁じます。

払込通知票

通常払込料金加入者負担

口座番号	東京 9 35087 8
加入者名	株式会社ビレッジセンター
金額	1720
払込人住所氏名	※（郵便番号　　　） TEL（　　　）
料金	払込み / 特殊
備考	
受付局日附印	

各欄の※印欄は、払込人において記載してください。

記載事項を訂正した場合は、その箇所に訂正印を押してください。

この払込通知票は、機械で使用しますので、下部の欄を汚さないよう特に御注意ください。また、本票を折り曲げたりしないでください。（郵政省）

切り取らないで郵便局にお出しください。

払込票

通常払込料金加入者負担

口座番号	東京 9 35087 8
加入者名	株式会社ビレッジセンター
金額	1720
払込人住所氏名	※
備考	
受付局日附印	

プログラムを収録しています。

■□■ ビレッジセンター出版案内 ■□■

ビレッジセンターの書籍は全国の書店にてお求めになれます。

各票の※印欄は、払込人において記載してください。

通常払込料金 加入者負担	払込通知票		
口座番号	東京 9-35087 8	金額	1800
加入者名	株式会社 ビレッジセンター	料金	払込み / 特殊 円
払込人住所氏名	※（郵便番号　　　） TEL（　　　）	備考	
		受付局日附印	

この払込通知票は、機械で使用しますので、下部の欄を汚さないよう特に御注意ください。また、本票を折り曲げたりしないでください。　（郵政省）

記載事項を訂正した場合は、その箇所に訂正印を押してください。

切り取らないで郵便局にお出しください。

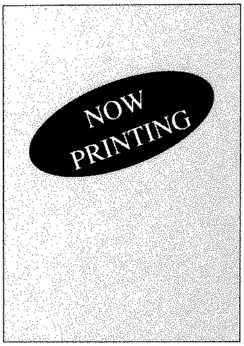
通常払込料金 加入者負担	払込票
口座番号	東京 9-35087 8
加入者名	株式会社 ビレッジセンター
金額	1800
払込人住所氏名	※
備考	受付局日附印

※本製品・マニュアルの一部または、全部を当社の許可なく複写、複製することを禁じます。

プリントアウトユーティリティ PRTシリーズ❷

レーザショット活用のための

PRT shot



豊嶋久道著
A5判 208頁
（3.5"＆5"FD付き）
定価3,200円

PRT++ の拡張ソフトとしてユーザの要望に応え誕生したPRT shotは、PRT++ の操作性を継承しつつレーザショット独自の機能をフル活用できます。オーバーレイ印刷の機能や、任意の文字サイズに対応、明朝体/ゴシック体の選択が可能、装飾文字の種類の増加など、多彩な機能がますます充実しています。本書は初心者の方にも分かりやすく丁寧に、上級者向けにはテクニック集として活用できる魅力いっぱいの一冊です。

ISBN4-938704-07-2

プリントアウトユーティリティ PRTシリーズ❶

PRT++ 100% 活用法

豊嶋久道著
A5判 208頁
（3.5"＆5"FD付き）
定価3,200円(本体3,107円)

テキスト印刷ユーティリティ PRT++初の解説書です。PRT++はパソコン通信を中心として配布されてきたシェアウェアで、その機能は高く評価されています。本書は、最新版Version 5.46のドキュメントリファレンスマニュアルに図版を取り入れ、理解しやすい構成となっているほか、より高度に使いこなすための活用編を加えてあります。また、付属のフロッピーディスクには、対応全機種のプログラムを収録しています。

ビレッジセンター出版案内

ビレッジセンターの書籍は全国の書店にてお求めになれます。

VILLAGE CENTER INC.
Kanda Amerex Bldg. 2F
1-16, 3-Chome, Misaki-Cho, Chiyoda-Ku Tokyo, 101 Japan